Continuous Control Artificial Potential Function Methods and Optimal Control

THESIS

R. Andrew Fields, Civ, USAF

AFIT-ENY-14-M-20

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENY-14-M-20

Continuous Control Artificial Potential Function Methods and Optimal Control

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

R. Andrew Fields, BSAE

Civ, USAF

March 2014

AFIT-ENY-14-M-20

Continuous Control Artificial Potential Function Methods and Optimal Control

R. Andrew Fields, BSAE
Civ, USAF

Approved:

//signed//	12 March 2014
Lt Col J. Agte, PhD, (Chairman)	date
//signed//	12 March 2014
R. Cobb, PhD (Member)	date
//signed//	12 March 2014
B. Liebst, PhD (Member)	date

Abstract

Artificial potential function methods (APFMs) are a class of computationally inexpensive control methods for driving a system to a desired goal while avoiding obstacles. Although APFMs have been applied successfully to a wide range of systems since the late 1980s, these control methods do have notable drawbacks. The general suboptimality of APFM results is one of these drawbacks, which is due to the fact that APFMs contain no cost function in their formulation.

This thesis first develops a new continuous control APFM for fully actuated systems called the Velocity Artificial Potential Function (VAPF) Method, which causes the system velocity to converge to the negative gradient of an artificial potential function. Then, methods for increasing APFM optimality are studied. First, an investigation is undertaken to determine if placing an APFM into an optimal control framework is a practical way of addressing the suboptimality of APFMs. While effective at increasing optimality of APFM results, this approach proves to be too computationally expensive to be practical. Finally, the Adaptive Artificial Potential Function developed by Muñoz is studied and implemented via the VAPF Method. This approach produces results with higher optimality than traditional APFMs but with negligibly greater computational expense.

To Mom and Dad

Acknowledgements

First and foremost, I would like to thank my research advisor, Lt. Col. Jeremy Agte. Without the countless conversations and whiteboard drawings in Lt. Col. Agte's office, this research never would have gotten off the ground. I'd also like to thank him for picking up the tab at all of our thesis lunches.

I would also like to express my gratitude to Dr. Josue Muñoz of the AFRL Space Vehicles Directorate, who was the thesis sponsor. Dr. Muñoz always eagerly provided his insight and technical expertise via telephone, email, and in-person meetings. His advice greatly influenced the research herein.

Finally, I would like to thank my thesis committee members, Dr. Richard Cobb (who also took the role of stand-in advisor during Lt. Col. Agte's deployment) and Dr. Bradley Liebst, for the great amount of time they took to review this manuscript. Their comments and suggestions markedly improved the thesis.

R. Andrew Fields

Table of Contents

Page

List of Figures

List of Tables

List of Abbreviations

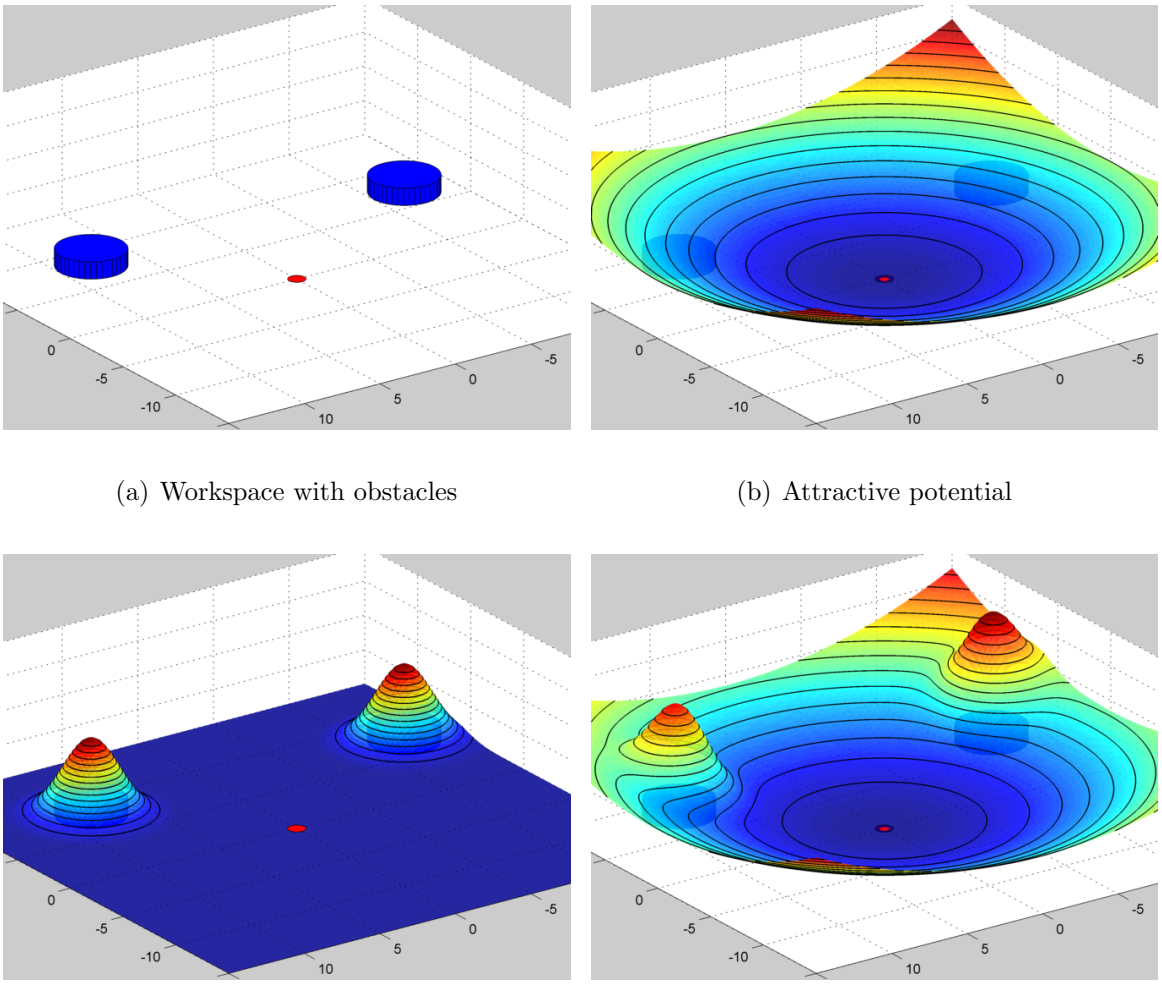
Continuous Control Artificial Potential Function Methods and Optimal Control

I. Introduction and Motivation

1.1 Artificial Potential Functions

An interesting problem in the field of guidance, navigation, and control (GNC) is how to drive a system from one state to another while avoiding certain regions, such as physical obstacles. One approach to solve such problems is the use of artificial potential function methods (APFMs), which have seen application in a wide variety of systems, including robotic manipulators, spacecraft, and autonomous road vehicles [1–3]. APFMs involve the creation of a scalar-valued, nonnegative artificial potential function (APF) in the system's workspace. The APF has its global minimum at the system's goal location. If the system follows a path of decreasing potential, then it will converge to the goal. Typically, an APF is composed from the superposition of two types of functions: attractive potentials and repulsive potentials. The attractive potential is a bowl- or well-shaped function (such a quadratic) whose global minimum is the goal point. Repulsive potentials are functions with large value near obstacles and small (or zero) value far from obstacles.

Figure 1.1 illustrates the construction of an APF. A planar workspace with two circular obstacles is depicted in Figure 1.1(a). Note that, for ease of viewing, the planar obstacles are shown as cylinders coming out of the plane, and the goal point is shown as a red dot. The attractive potential is shown in Figure 1.1(b); a quadratic attractive potential is used here. The repulsive potentials are depicted in Figure 1.1(c); Gaussian functions are used in this instance. Finally, Figure 1.1(d) shows the complete APF formed by the superposition of the attractive and repulsive potentials.



(a) Workspace with obstacles

(b) Attractive potential

(c) Repulsive potential

(d) Resulting APF

Figure 1.1: Construction of an APF

Typically, APFMs attempt to drive the system along the negative gradient direction of the APF. An example of such a scheme for a vehicle with impulsive control is given here: A vehicle proceeds from an initial condition under its own dynamics. At every time step, the APF gradient is calculated at the vehicle's location. If the angle between the vehicle's velocity vector and the negative of the APF gradient is larger than some tolerance, then a control input is generated to make the vehicle's velocity equal to the negative of the APF gradient. Otherwise, the vehicle proceeds without any control input. Thus, the path of the vehicle is analogous to the steepest descent on the potential surface. As seen in this scheme, the AFPM requires no path planning and relies only on the current APF information.

One significant advantage of APFMs is their computational efficiency. Because the APF itself is created from analytical expressions, such as Gaussians and quadratics, the resulting control commands are also analytical expressions. Even if the workspace contains many obstacles, the control commands to avoid those obstacles and converge to the goal are analytical.

There are two major drawbacks to AFPMs. The first is the existence of spurious local minima in the APF. A vehicle conducting a steepest descent on the APF can become stuck in the local minima, and will not converge to the goal. The local minima exist in locations where the gradients of the attractive potential and repulsive potentials add to zero. This thesis does not investigate the challenges of local minima and will consider only APFs without them.

The second drawback is the suboptimal performance from APFMs. The APF formulation does not incorporate any cost function (such as time, fuel consumption, etc.) and thus the system trajectories are generally suboptimal with respect to such cost functions. Improvement of APFM performance is a major area of investigation for this thesis.

1.2 Optimal Control and APFs

While APFMs provide computationally efficient control schemes for traveling to a goal and simultaneously avoiding obstacles, an area of improvement should be to improve their optimality with respect to some cost function. In addition, it is desired that such optimizations should not diminish the computational advantage of APFMs. In theory, the optimization of an APFM could be less computationally expensive than solving an optimal control problem with obstacle avoidance path constraints. Consider the following general optimal control problem for a system that must minimize a cost function J while avoiding obstacles (represented by circles in two dimensions):

$$\begin{aligned}
\min_{\boldsymbol{u}} \quad & J = \Phi\left(\boldsymbol{x}_f\right) + \int_0^{t_f} L\left(\boldsymbol{x}, \boldsymbol{u}\right) \, \mathrm{d}t \\
\text{subject to} \quad & \dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \\
& R_j - \|\boldsymbol{r}(t) - \boldsymbol{o}_j\| \leq 0 \qquad \text{for } j = 1 \text{ to } l, \\
& \boldsymbol{x}(0) = \boldsymbol{x}_0 \\
& \psi\left(\boldsymbol{x}(t_f)\right) = 0
\end{aligned}$$

where $\boldsymbol{x}$ is the state vector, $\boldsymbol{u}$ is the control vector, R_j is the radius of the j^{th} obstacle, $\boldsymbol{r}(t)$ is the system's position vector, $\boldsymbol{o}_j$ is the position of the center of the j^{th} obstacle, l is the number of obstacles, and ψ is a terminal constraint. The inclusion of the obstacle avoidance path constraints adds significantly to the computational expense of solving such a problem because of the Lagrange multipliers associated with the path constraints, along with the fact that they are non-convex constraints.

Now consider that the system is controlled via an APFM, and the control law from the APFM is represented by

$$\boldsymbol{u}_{APF} \equiv \boldsymbol{u}_{APF}\left(\boldsymbol{x}, \mathit{APFM\ parameters}\right), \tag{1.1}$$

where *APFM parameters* may be shaping parameters that determine the shape of the APF and/or control parameters used in the APFM. With this control law

$\boldsymbol{u}_{APF}$, the system automatically avoids obstacles, and thus the path constraints in the optimal control problem may be removed. The independent variables for the optimal control problem are now *APFM parameters*. That is, the optimal control problem becomes

$$\begin{aligned}
\min_{APFM\ parameters} \quad & J = \Phi\left(\boldsymbol{x}_f\right) + \int_0^{t_f} L\left(\boldsymbol{x}, APFM\ parameters\right)\,\mathrm{d}t \\
\text{subject to} \quad & \dot{\boldsymbol{x}} = f(\boldsymbol{x}, APFM\ parameters) \\
& \text{bounds on } APFM\ parameters. \\
& \boldsymbol{x}(0) = \boldsymbol{x}_0 \\
& \psi\left(\boldsymbol{x}(t_f)\right) = 0
\end{aligned}$$

The non-convex path constraints have been removed, and bounds on the *APFM parameters* have been added. These constraints (oftentimes a positivity requirement) are convex and are handled by numerical schemes with much less computational expense than non-convex path constraints. Thus, it could be more computationally efficient to solve an APFM optimization than a regular optimal control problem with obstacle avoidance path constraints.

It should be noted that the optimum cost from solving the regular optimal control problem may be impossible to achieve from an APFM optimization. This is because APFMs typically allow only certain directions of motion (e.g. moving in a direction of decreasing potential). Nonetheless, the possibility of computational savings warrants investigation into APFM optimization.

1.3 Contributions to the Field

This thesis contributes to the field of APFMs in several ways. First, a new continuous control APFM known as the Velocity Artificial Potential Function Method is developed in Chapter III. The method is the first (to this author's knowledge) continuous control APFM which causes a system's velocity to converge exponentially

to the negative gradient of the APF. Then, an investigation is undertaken to determine if placing an APFM into an optimal control framework is a practical way of addressing the suboptimality of APFMs. The results indicate that this approach, while effective at increasing APFM optimality, is impractical due to the large computational expense involved. Finally, the Adaptive Artificial Potential Function developed by Muñoz [4] is combined with the Velocity Artificial Potential Function Method to produce results with higher optimality than traditional APFMs but negligibly greater computational expense.

1.4 Thesis Outline

This chapter has introduced the notion of APFMs and the optimization thereof. Chapter II presents background information on AFPMs and a survey of literature in the field. Chapter III details the development of several continuous control APFMs that are applicable to fully actuated systems. A new method called the Velocity Artificial Potential Function Method is developed and demonstrated in an example application. Chapter IV examines ways of increasing the optimality of APFM results. First, an APFM is placed into an optimal control framework and is then evaluated on four test cases. Then, an Adaptive Artificial Potential Function is implemented via the control method developed in Chapter III. Key conclusions are discussed in Chapter V.

II. Background and Literature Review

This chapter presents background information on AFPMs and reviews relevant literature in the APFM field. First, the various areas in which APFMs have been applied are discussed. Then, the advantages and drawbacks of APFMs are presented, followed by previous efforts aimed at addressing the drawbacks. After establishing this background on APFMs, the research scope for this thesis is presented.

2.1 APFM Application Areas

APFMs have seen application in diverse fields of study. Their earliest usage was in control of robotic manipulators. In fact, it was Khatib who developed the first APFM during his doctoral work on obstacle avoidance for manipulators [5]. Khatib modeled the APF negative gradient as a force acting on the links of the manipulator [1]. These artificial forces were then translated into torque inputs to the joint actuators. APFMs have been applied to numerous other aspects of manipulator control, including tentacle manipulators [6] and human-manipulator interaction [7].

APFMs have also been applied extensively to spacecraft rendezvous and formation flight. In the 1990s, Lopez and McInnes introduced an APFM where the APF itself was considered to be a Lyapunov function.[1] In this method, the chase vehicle proceeded under its own open-loop dynamics (the Clohessy-Wiltshire equations[2]) until the time rate of change of potential became nonnegative. At that time, a thrust impulse was applied to make the chase vehicle's velocity a scalar multiple of the negative gradient of the APF. In effect, the method interpreted the negative APF gradient as a desired velocity field, and corrective impulses were applied whenever the chase vehicle velocity deviated from the negative gradient direction by $\pi/2$ radians. A separate terminal guidance routine was used to ensure docking along a specific axis. Other spacecraft control papers have implemented continuous control APFMs which interpret the negative APF gradient as an acceleration field, and damping is applied

[1] Lyapunov functions are discussed in Section 3.2.
[2] The Clohessy-Wiltshire equations are introduced in Section 3.5.

to eliminate oscillation around the goal point [8, 9]. Such a method is described in detail in Section 3.2.

APFMs have been applied in various other fields where motion planning or guidance is required. In [10], Roussos et al. used an APFM as a low-level controller within a model predictive control scheme. The authors successfully applied their method to a conflicted air traffic control scenario with wind uncertainty. In [11], Zhenhai et al. investigated the use of an APFM to generate preview trajectories for an automatic lanekeeping system in autonomous road vehicles. Areas of application outside of engineering exist as well. For instance, Ulutas et al. used an APFM to study how protein molecules fold to arrive at their equilibrium configuration [12].

This section has reviewed a few of the many areas in which APFMs have been successfully applied. Their computational efficiency and ease of accounting for obstacles have made APFMs an attractive option in almost any field where motion planning and guidance are required.

2.2 Advantages of APFMs

The primary advantage of APFMs is that they provide a computationally efficient scheme for driving a system to its desired goal while avoiding collisions with obstacles. This computational efficiency is achieved because the required control inputs are obtained from analytical expressions. This attribute makes APFMs attractive for use on systems where onboard computational power is constrained, such as in small spacecraft.

An example will be presented here to demonstrate the simplicity of APFM control laws. This example comes from Lopez and McInnes [2], who wrote one of the earliest papers that applied an APFM to spacecraft rendezvous. For the construction of the APF, ϕ, the authors chose a quadratic function for the attractive potential:

$$\phi_{\text{att}}(\boldsymbol{r}) = \boldsymbol{r}^{\intercal}\mathbf{M}\boldsymbol{r}, \tag{2.1}$$

where $\boldsymbol{r}$ is the chase vehicle position vector, and $\mathbf{M}$ is a positive definite matrix. The repulsive potentials, which impose areas of high potential near obstacles, were chosen as Gaussian functions:

$$\phi_{\text{rep},i}(\boldsymbol{r}) = \lambda_i \exp\left(-\sigma_i^{-1}(\boldsymbol{r} - \boldsymbol{r}_{obs,i})^\intercal \mathbf{N}_i(\boldsymbol{r} - \boldsymbol{r}_{obs,i})\right), \tag{2.2}$$

where $\boldsymbol{r}_{obs,i}$ is the position of the center of the i^{th} obstacle, and λ_i, σ_i, and $\mathbf{N}_i$ represent different shaping parameters chosen by the designer. The complete ϕ was then

$$\phi = \phi_{\text{att}} + \sum_{i=1}^{m} \phi_{\text{rep},i}, \tag{2.3}$$

where m is the number of obstacles.

To obtain the impulsive control law, ϕ was interpreted as a Lyapunov function, and the condition for asymptotic convergence to the goal (the origin) was that

$$\dot{\phi}(\boldsymbol{r}, \dot{\boldsymbol{r}}) < 0 \quad \forall \boldsymbol{r} \neq \mathbf{0}. \tag{2.4}$$

Whenever $\dot{\phi} < 0$ was satisfied, the chase vehicle was allowed to drift under its own dynamics. Then, whenever $\dot{\phi}$ vanished, an impulsive velocity change was effected to maintain $\dot{\phi} < 0$. The impulsive velocity change was

$$\begin{aligned}\Delta\dot{\boldsymbol{r}} &= -k\nabla\phi - \dot{\boldsymbol{r}} \\ &= -2k\mathbf{M}\boldsymbol{r} + 2k\sum_{i=1}^{m} \lambda_i\sigma_i^{-1} \exp\left(-\sigma_i^{-1}(\boldsymbol{r} - \boldsymbol{r}_{obs,i})^\intercal \mathbf{N}_i(\boldsymbol{r} - \boldsymbol{r}_{obs,i})\right) \mathbf{N}_i(\boldsymbol{r} - \boldsymbol{r}_{obs,i}) - \dot{\boldsymbol{r}},\end{aligned} \tag{2.5}$$

where k is a constant positive scalar. In effect, this impulsive control law corrects the velocity of the chase vehicle to be a scalar multiple of $-\nabla\phi$ whenever $\dot{\phi}$ vanishes. It can be seen that the required impulse in (2.5) is an analytic expression that can be evaluated in real time even with modest computational resources.

2.3 Drawbacks of APFMs

One of the major drawbacks of APFMs is the existence of local minima that can cause a vehicle to become stuck away from its goal. These local minima are created in locations where

$$\nabla\phi_{\text{att}} = -\nabla\phi_{\text{rep}}. \tag{2.6}$$

A common source of local minima is two or more obstacles that are in close proximity to each other, as depicted in Figure 2.1. A single obstacle also creates a saddle point

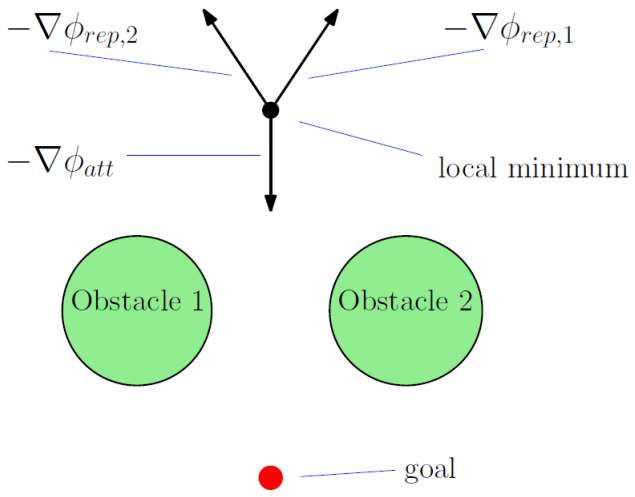


Figure 2.1: Typical Local Minimum Caused by Two Obstacles in Close Proximity

along the line between the obstacle and the goal. Saddle points, however, are of less concern than local minima because they are unstable equilibria, and so vehicles are much less likely to become trapped there. Several attempts to address the local minimum problem are addressed in Section 2.4.

Another significant drawback of APFMs is their suboptimal performance. APFM formulations typically do not incorporate any cost function; APFM trajectories are the result of the shape of the APF itself and any control parameters used. In most cases the APF shaping parameters and control parameters are fixed values which must be chosen based upon the designer's experience. APFMs can also often cause abrupt initial accelerations and obstacle avoidance maneuvers that give excessively

wide berth. Additionally, the shape of the APF is not reflective of the system's underlying dynamics. Therefore, gradient-following trajectories on static APFs cannot take advantage of any underlying dynamics that could help to minimize a cost function such as control. Section 2.5 describes several efforts aimed at increasing the optimality of APFMs.

2.4 Methods Addressing the Local Minimum Problem

Various researchers have investigated solutions to the local minimum problem. In [13], Vadakkepat et al. introduced an algorithm that imposed an additional "escape force" upon the vehicle whenever the APF gradient fell below certain thresholds. While the escape force algorithm proved effective for avoiding local minima, the threshold values and escape force parameters were chosen by a genetic algorithm[3], thereby reducing some of the simplicity which makes APFMs attractive.

In [9], Ahsun proposed a local minimum avoidance algorithm for spacecraft in formation flight maneuvers. First, the locations of all local minima in the workspace had to be computed. Then, whenever a spacecraft came within some threshold distance ϵ of a local minimum, an additional acceleration perpendicular to the spacecraft's velocity was commanded. This algorithm was proven to provide asymptotic convergence to the desired goal location even when local minima were present. However, the algorithm required pre-calculation of all local minimum locations, which can be very computationally intensive for complicated geometries or geometries with moving obstacles.

In [14], Rimon and Koditschek derived a specialized form of APF known as a navigation function, which is devoid of local minima except for the desired goal position. All other critical points on the navigation function are unstable saddle points. The special construction proposed by Rimon and Koditschek is remarkable in that it can be applied to any generalized sphere world, thereby providing a feasible

[3] A genetic algorithm is a heuristic method for solving optimization problems based on the process of natural selection.

solution to the robot navigation problem for any geometry where the obstacle shapes can be mapped to n-spheres by a diffeomorphism.[4] However, navigation functions can be difficult to compute. In addition, navigation functions can possess gradients that change rapidly from extremely shallow to extremely steep, resulting in velocity profiles that may be undesirable.

While the methods above each provide means of circumventing the local minimum problem, they also introduce other drawbacks that detract from the original simplicity of APFMs. The local minimum problem is still an area of active study for APFM researchers.

2.5 Methods Addressing the Suboptimality Problem

The section describes several research efforts that addressed the drawback of suboptimal APFM trajectories. All of the methods hinged upon modifying the shape of the APF itself to create a trajectory with a higher degree of optimality with respect to some cost function.

Vadakkepat et al. [13] examined the use of a genetic algorithm to vary the repulsive potential shaping parameters of an APF to achieve optimal results. The cost function was a sum of three geometric factors: goal factor, obstacle factor, and minimum path length factor. An optimal trajectory would reach the goal while following the shortest collision-free path. The genetic algorithm successfully varied the repulsive potential shaping factors through time to achieve trajectories that were more optimal than those obtained with a static APF.

In his research, Ahsun [9] noted the problem of large goal-convergence times for APFM trajectories. This slow convergence occurs because the gradient of the APF approaches zero as the system approaches the goal, and therefore the system velocity tends to zero. As a result, systems can spend a large amount of time in the vicinity of the goal before convergence criteria are met. To remedy this problem, Ahsun intro-

[4] A diffeomorphism is a smooth, differentiable mapping.

duced Potential Function Shaping, whereby additional terms are added to the APF which steepen the gradient only in the vicinity of the goal. With this simple modification, the goal-convergence times in Ahsun's satellite formation maneuver simulations were greatly reduced. Potential Function Shaping added no additional computational expense since only a static term was added to the APF.

Muñoz also sought to address the suboptimalty of APFM trajectories, specifically in the case of propellant consumption during close range spacecraft rendezvous [4]. Muñoz noted that gradient-following trajectories generated by APFMs did not reflect the underlying system dynamics, but rather only the geometrical shape of the APF itself. In the simple case of an APF with no obstacles, a gradient-following trajectory would follow a straight line toward the goal. The fact that the shape of the APF had no relation to the dynamics of the system meant that APFM trajectories could not take advantage of the underlying dynamics of the system to achieve lower propellant consumption during rendezvous. Muñoz also noted the problem of large convergence times. To address these problems, Muñoz devised an Adaptive Artificial Potential Function (AAPF) where the gradient of the attractive potential would adapt with time to match a prescribed velocity profile. By setting a desired two-burn rendezvous as the prescribed velocity profile, Muñoz achieved both lower propellant consumption and shorter convergence times compared to cases with static APFs. It should be noted that only the attractive potential was adapting in the AAPF scheme; the repulsive potentials for obstacles remained constant. That is, the adaptive attractive potential allowed the spacecraft to follow the prescribed velocity profile when far away from obstacles; the static repulsive potentials caused the vehicle to deviate from the prescribed velocity profile in order to avoid collisions. The AAPF scheme added very little computational expense because the adapting shaping parameters were obtained through a simple forward propagation. The AAPF methodology will be examined in greater detail in Section 4.5.

While the methods described in this section successfully increased APFM optimality in their respective applications, they also possess limitations. In the work of

Vadakkepat et al., the cost functions that are minimized are only geometric in nature and are not related to the system dynamics. In addition, their methodology has no means of imposing boundary conditions or transfer times on the trajectories. The use of a genetic algorithm also increases the computational complexity of the problem. Ahsun's Potential Function Shaping, while successful at decreasing convergence time in the vicinity of the goal, is somewhat limited otherwise. The method is not generally applicable to the minimization of other cost functions, such as control usage. In addition, parameters for the additional APF terms must be chosen based only upon the designer's experience. Regarding Muñoz's AAPF scheme, one limitation is that its increased optimality hinges upon the prescription of a reference velocity profile with optimal characteristics. Such a desired velocity profile may be unknown for complex problems. Additionally, because only the attractive potential is adaptive, the increase in optimality may be small for problem geometries with strongly influential obstacles.

2.6 Scope of Work

The first task of this research is to develop, in Chapter III, a continuous-control APFM for fully actuated systems. This APFM should cause the system velocity to converge to the negative gradient of the APF. To this author's knowledge, no such continuous control scheme has been previously developed, although related impulsive control schemes do exist (e.g. as shown in Section 2.2). This continuous control scheme will be illustrated in an idealized spacecraft rendezvous scenario.

The second half of the research, presented in Chapter IV, will investigate the possibility of increasing the optimality of the APFMs while maintaining their most attractive feature - computational efficiency. Figure 2.2 depicts the research space for APFM optimality. At the bottom left are located traditional, static APFMs with low optimality but low computational expense. On the other hand, optimal control solutions provide high optimality at large computational cost. The promising middle ground between these two extremes warrants investigation.

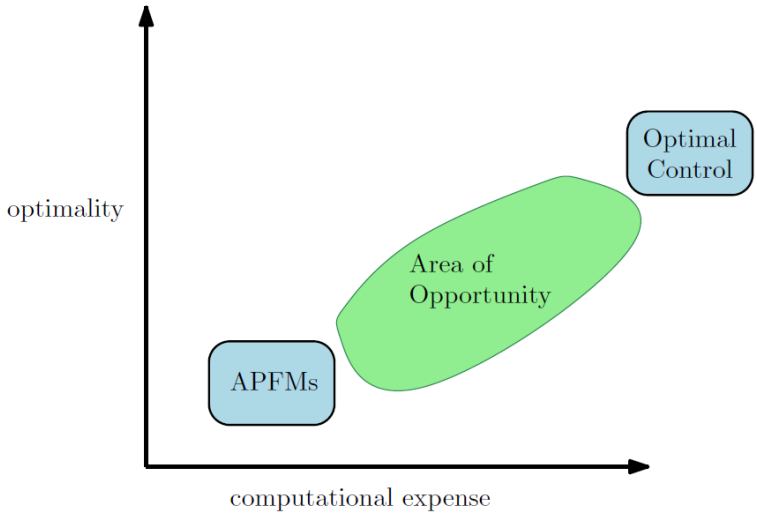


Figure 2.2: Research Space for APFM Optimality

This investigation will be accomplished by transcribing the APFM directly into an optimal control framework. Once in the optimal control framework, control parameters within the APFM will serve as the independent variables to be solved by the optimal control software. There is a possibility of computational savings in using this approach because obstacle-avoidance path constraints in the problem statement are removed. Additionally, as an alternative to the optimal control framework, the AAPF of Muñoz will be combined with the continuous control method from Chapter III and applied in an example scenario.

III. Continuous Control Method Development

This chapter details the development of several continuous control APFMs for fully actuated systems. It is desired to obtain a method which causes the system's velocity to be the negative gradient of the APF. Several methods are examined, ultimately leading to the Velocity Artificial Potential Function Method in Section 3.3. A brief discussion on APF construction is then followed by an example application of the Velocity Artificial Potential Function Method.

3.1 Method 1

This section presents Method 1, the first of three continuous control methods in this chapter. Method 1 is a first-glance attempt at formulating a continuous control law which causes a system's velocity to equal $-\nabla\phi$. All of the methods investigated in this chapter are first developed on a simple problem with one-dimensional double integrator dynamics and a quadratic ϕ. Candidate methods deemed promising are then generalized for application to other dynamics and forms of ϕ. The state-space representation for the one-dimensional double integrator dynamics is

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u, \tag{3.1}$$

where m is the system mass. The APF used for the one-dimensional problem is

$$\phi = \frac{1}{2} M x^2, \tag{3.2}$$

where M is a positive definite shaping factor.

Method 1 begins with the premise that the system's velocity should be equal to the negative gradient of ϕ:

$$\begin{aligned} \dot{x} &= -\nabla\phi \\ &= -Mx. \end{aligned} \tag{3.3}$$

Taking the time derivative of (3.3) gives

$$\ddot{x} = -M\dot{x}. \tag{3.4}$$

Substituting (3.4) into (3.1) and solving for u yields

$$u = -Mm\dot{x}. \tag{3.5}$$

Substitution of (3.5) into (3.1) closes the loop and yields

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ 0 & -M \end{bmatrix} \boldsymbol{x}. \tag{3.6}$$

The eigenvalues for this closed-loop system are $\lambda = 0$ and $\lambda = -M$, so the system is stable.

The analytical solution for the system in (3.6) is

$$x(t) = x_0 + \frac{\dot{x}_0}{M} - \frac{\dot{x}_0}{M}\exp(-Mt). \tag{3.7}$$

From inspection of the analytical solution, one can observe that the position will converge to the origin only if $\dot{x}_0 = -Mx_0$. That is, only if the initial velocity exactly matches $-\nabla\phi$ at the initial position. Thus, Method 1 is not effective at driving the system to the goal when the system has an arbitrary initial state. The deficiency of Method 1 comes from the inherent assumption in (3.3) that the velocity is equal to $-\nabla\phi$, rather than specifying $-\nabla\phi$ as a desired velocity. Method 1 was not pursued further.

3.2 Lyapunov Method

3.2.1 Development on One-dimensional Problem. The Lyapunov Method, which has been used in previous APFM-related works [8,9,15], combines the APF with Lyapunov theory. Lyapunov theory gives a sufficient condition for the stability of a

dynamic system [16]. Given a system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$, a Lyapunov Candidate Function $V(\boldsymbol{x})$ is defined as a scalar valued function that is positive definite and radially unbounded, i.e.

$$\begin{aligned} &V(\mathbf{0}) = 0 \\ &V(\boldsymbol{x}) > 0 \quad \forall \boldsymbol{x} \neq \mathbf{0} \\ &|\boldsymbol{x}| \to \infty \Rightarrow V(\boldsymbol{x}) \to \infty. \end{aligned}$$

The system is asymptotically stable if the time derivative of V is negative definite, i.e.

$$\dot{V}(\boldsymbol{x}) < 0 \quad \forall \boldsymbol{x} \neq \mathbf{0}. \tag{3.8}$$

If $\dot{V}(\boldsymbol{x})$ is only negative semidefinite, then the system is stable. In some cases, LaSalle's Invariance Principle may be used to show asymptotic stability even when $\dot{V}(\boldsymbol{x})$ is only negative semidefinite [17]. Again, these conditions are sufficient, but not necessary, for system stability.

The Lyapunov Method creates a Lyapunov Candidate Function (LCF) which incorporates the APF, ϕ. In this method, an acceleration which causes $\dot{V}$ to be negative semidefinite is chosen. Using the one-dimensional double integrator dynamics and quadratic potential, (3.1) and (3.2), an LCF, V, is defined as

$$\begin{aligned} V(x, \dot{x}) &= \frac{1}{2} P \dot{x}^2 + \phi \\ &= \frac{1}{2} P \dot{x}^2 + \frac{1}{2} M x^2, \end{aligned} \tag{3.9}$$

where P is a positive scalar. Differentiating V with respect to time yields

$$\dot{V}(x, \dot{x}) = P \dot{x} \ddot{x} + M x \dot{x}. \tag{3.10}$$

To make $\dot{V}$ negative semidefinite, the acceleration is chosen as

$$\ddot{x} = -\frac{1}{P}\left(K_A\dot{x} + Mx\right), \tag{3.11}$$

where K_A is a positive scalar. Substitution of this acceleration into (3.10) gives

$$\begin{aligned}\dot{V}\left(x,\dot{x}\right) &= P\dot{x}\left(-\frac{1}{P}\left(K_A\dot{x} + Mx\right)\right) + Mx\dot{x} \\ &= -K_A\dot{x}^2,\end{aligned} \tag{3.12}$$

which is negative semidefinite. Therefore the system is stable. Additional analysis using LaSalle's Invariance Principle can be used to show that the system is, in fact, asymptotically stable. In this simple case, linear analysis also proves asymptotic stability.

Taking the acceleration expression (3.11) and substituting into the dynamics (3.1) and then solving for u yields

$$u = -\frac{m}{P}\left(K_A\dot{x} + Mx\right). \tag{3.13}$$

Substitution of this control into the dynamics closes the loop and yields

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ -\frac{M}{P} & -\frac{K_A}{P} \end{bmatrix}\boldsymbol{x}. \tag{3.14}$$

The eigenvalues of the closed-loop system are $\lambda = -\frac{K_A}{2P} \pm \sqrt{\left(\frac{K_A}{2P}\right)^2 - \frac{M}{P}}$. Since the eigenvalues have a negative real part, the system is indeed asymptotically stable. The following subsection generalizes the Lyapunov Method to higher dimensions, arbitrary ϕ, and arbitrary fully actuated dynamics.

3.2.2 Generalization of the Lyapunov Method. Consider an arbitrary APF ϕ in an n-dimensional space applied to a system with arbitrary second-order, fully

actuated dynamics. It is desired to drive the system asymptotically to the origin. In a similar manner to (3.9), an LCF is defined as

$$V(\boldsymbol{r}, \dot{\boldsymbol{r}}) = \frac{1}{2}\dot{\boldsymbol{r}}^{\intercal}\mathbf{P}\dot{\boldsymbol{r}} + \phi, \tag{3.15}$$

where $\mathbf{P}$ is a positive definite matrix. The time derivative of V is

$$\dot{V}(\boldsymbol{r}, \dot{\boldsymbol{r}}) = \dot{\boldsymbol{r}}^{\intercal}\mathbf{P}\ddot{\boldsymbol{r}} + \dot{\boldsymbol{r}}^{\intercal}\nabla\phi. \tag{3.16}$$

In order to make $\dot{V}$ negative semidefinite, the acceleration is chosen as

$$\ddot{\boldsymbol{r}} = -\mathbf{P}^{-1}\left(\mathbf{K}_A\dot{\boldsymbol{r}} + \nabla\phi\right), \tag{3.17}$$

where $\mathbf{K}_A$ is a positive definite matrix. Substitution of this acceleration into (3.16) yields

$$\begin{aligned}\dot{V}(\boldsymbol{r}, \dot{\boldsymbol{r}}) &= -\dot{\boldsymbol{r}}^{\intercal}\mathbf{P}\mathbf{P}^{-1}\left(\mathbf{K}_A\dot{\boldsymbol{r}} + \nabla\phi\right) + \dot{\boldsymbol{r}}^{\intercal}\nabla\phi \\ &= -\dot{\boldsymbol{r}}^{\intercal}\mathbf{K}_A\dot{\boldsymbol{r}} - \dot{\boldsymbol{r}}^{\intercal}\nabla\phi + \dot{\boldsymbol{r}}^{\intercal}\nabla\phi \\ &= -\dot{\boldsymbol{r}}^{\intercal}\mathbf{K}_A\dot{\boldsymbol{r}}, \end{aligned} \tag{3.18}$$

which is negative semidefinite and thus the system is stable. If ϕ is free of spurious local minima, then LaSalle's Invariance Principle can be used to prove asymptotic stability except for a thin set of initial conditions that drive the system to saddle points of ϕ (an inherent feature in any ϕ which includes obstacles) [9]. Thus, (3.17) defines an acceleration that drives the system to the goal and ensures that V is always nonincreasing. Examination of (3.17) shows that the Lyapunov Method essentially interprets $-\mathbf{P}^{-1}\nabla\phi$ as a prescribed acceleration field (rather than a velocity field) to which a damping acceleration is applied. A sufficient, but not necessary condition for obstacle collision avoidance is that the value of ϕ at obstacle boundaries must be

greater than the initial value of V, i.e.

$$\phi|_{\text{obstacle boundaries}} \geq V_0 = \frac{1}{2}\dot{\boldsymbol{r}}_0^{\intercal}\mathbf{P}\dot{\boldsymbol{r}}_0 + \phi_0. \tag{3.19}$$

3.3 Velocity Artificial Potential Function Method

3.3.1 Development on One-dimensional Problem. While the Lyapunov Method is effective at driving a system to its goal, one possible shortcoming of the method is that it does not cause the system's velocity to equal $-\nabla\phi$. The Velocity Artificial Potential Function (VAPF) Method was developed to drive a system with velocity $-\nabla\phi$. The concept of the VAPF Method is to define a second potential function, called the velocity artificial potential function, in the velocity space of the vehicle. The VAPF, represented by ϕ_v, has its desired goal (i.e. the minimum) located at $-\nabla\phi$. Then an acceleration is solved that causes the time derivative of the VAPF to be negative semidefinite. This ensures that the system velocity converges to $-\nabla\phi$.

The VAPF, ϕ_v, is defined as

$$\begin{aligned}\phi_v &= \frac{1}{2}(\dot{x}_{\text{desired}} - \dot{x})^2 \\ &= \frac{1}{2}(-\nabla\phi - \dot{x})^2 \\ &= \frac{1}{2}(Mx + \dot{x})^2.\end{aligned} \tag{3.20}$$

Taking the time derivative of (3.20) gives

$$\dot{\phi}_v = (Mx + \dot{x})(M\dot{x} + \ddot{x}). \tag{3.21}$$

An acceleration must now be chosen that causes $\dot{\phi}_v$ to be negative semidefinite. Moreover, it is desirable that the acceleration cause ϕ_v to decay exponentially. Such an acceleration would cause the system's velocity to converge exponentially to $-\nabla\phi$. The

acceleration is chosen as

$$\ddot{x} = -M\dot{x} - K_d(Mx + \dot{x}), \tag{3.22}$$

where K_d is a positive scalar gain. Substituting (3.22) into (3.21) yields

$$\begin{aligned}
\dot{\phi}_v &= (Mx + \dot{x})(M\dot{x} + (-M\dot{x} - K_d(Mx + \dot{x}))) \\
&= K_d(Mx + \dot{x})(-Mx - \dot{x}) \\
&= -K_d(Mx + \dot{x})^2 \\
&= -2K_d\phi_v,
\end{aligned} \tag{3.23}$$

which is the time derivative of a decaying exponential. Thus, ϕ_v may be expressed as function of time as

$$\phi_v(t) = \phi_{v,0} \exp(-2K_d t). \tag{3.24}$$

And therefore the square of the difference between the desired velocity ($-\nabla\phi$) and the actual velocity exponentionally converges to zero.

The VAPF Method acceleration in (3.22) can be used in conjunction with the dynamics (3.1) to solve for u:

$$u = -m(K_d M x + (K_d + M)\dot{x}). \tag{3.25}$$

Closing the loop with this control law gives

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ -K_d M & -(M + K_d) \end{bmatrix} \boldsymbol{x}. \tag{3.26}$$

The resulting closed-loop eigenvalues are $\lambda = -M$, $\lambda = -K_d$. Because the eigenvalues are negative definite, the system converges asymptotically to the origin. Thus, VAPF Method inverts the system dynamics to drive to velocity to $-\nabla\phi$. The following subsection generalizes the VAPF Method.

3.3.2 Generalization of the VAPF Method. Consider an arbitrary APF ϕ in an n-dimensional space applied to a system with arbitrary second-order, fully actuated dynamics. It is desired that the system's velocity, $\dot{\boldsymbol{r}}$, be equal to $-\nabla\phi$. Define the VAPF as

$$\phi_v = \frac{1}{2}(\nabla\phi + \dot{\boldsymbol{r}})^\intercal(\nabla\phi + \dot{\boldsymbol{r}}). \tag{3.27}$$

Note that (3.27) may expanded as

$$\begin{aligned}\phi_v &= \frac{1}{2}(\nabla\phi_1 + \dot{r}_1)^2 + \frac{1}{2}(\nabla\phi_2 + \dot{r}_2)^2 + \cdots + \frac{1}{2}(\nabla\phi_n + \dot{r}_n)^2 &(3.28)\\ &= \phi_{v_1} + \phi_{v_2} + \cdots + \phi_{v_n}, &(3.29)\end{aligned}$$

where $\phi_{v,i}$ is the specific VAPF for the i^{th} component of velocity. The time derivative of (3.27) is

$$\dot{\phi}_v = (\nabla\phi + \dot{\boldsymbol{r}})^\intercal(H(\phi)\dot{\boldsymbol{r}} + \ddot{\boldsymbol{r}}), \tag{3.30}$$

where $H(\phi)$ is the Hessian of ϕ. Now $\ddot{\boldsymbol{r}}$ must be chosen to cause $\dot{\phi}_v$ to be negative semidefinite. Let

$$\ddot{\boldsymbol{r}} = -H(\phi)\dot{\boldsymbol{r}} - \mathbf{K}_d(\nabla\phi + \dot{\boldsymbol{r}}) \tag{3.31}$$

where $\mathbf{K}_d$ is a diagonal, positive definite matrix whose i^{th} diagonal entry is K_{d_i}. Substitution of this acceleration expression into (3.30) yields

$$\begin{aligned}\dot{\phi}_v &= (\nabla\phi + \dot{\boldsymbol{r}})^\intercal(H(\phi)\dot{\boldsymbol{r}} + (-H(\phi)\dot{\boldsymbol{r}} - \mathbf{K_d}(\nabla\phi + \dot{\boldsymbol{r}})))\\ &= (\nabla\phi + \dot{\boldsymbol{r}})^\intercal\mathbf{K_d}(-\nabla\phi - \dot{\boldsymbol{r}})\\ &= -2\left(K_{d_1}\phi_{v_1} + K_{d_2}\phi_{v_2} + \cdots + K_{d_n}\phi_{v_n}\right) &(3.32)\end{aligned}$$

which is negative semidefinite by the definitions of ϕ_{v_i} and K_{d_i}. Given this time derivative, ϕ_v may be expressed as a function of time as

$$\phi_v(t) = \phi_{v_1,0}\exp(-2K_{d_1}t) + \phi_{v_2,0}\exp(-2K_{d_2}t) + \cdots + \phi_{v_n,0}\exp(-2K_{d_n}t). \tag{3.33}$$

Note that in the VAPF Method, an acceleration (3.31) has been defined which causes the 2-norm of the difference between $-\nabla\phi$ and the system velocity to exponentially converge to zero. Because of the assumption of a fully actuated system, this acceleration is always achievable.

If the system dynamics are linear, the required control to achieve the VAPF Method acceleration can be easily computed. Consider the dynamics of the fully actuated, second-order linear system:

$$\dot{\boldsymbol{x}} = \left[\begin{array}{c|c} \mathbf{0} & \mathbf{I} \\ \hline \mathbf{A}_{acc} & \end{array}\right] \boldsymbol{x} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{acc} \end{bmatrix} \boldsymbol{u}, \tag{3.34}$$

where $\mathbf{A}_{acc}$ is the last n rows of the $\mathbf{A}$ matrix, and $\mathbf{B}_{acc}$ is the last n rows of the $\mathbf{B}$ matrix. The state vector $\boldsymbol{x}$ is defined as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r} \\ \dot{\boldsymbol{r}} \end{bmatrix}.$$

Substitution of the VAPF Method acceleration from (3.31) into (3.34) and solving for $\boldsymbol{u}$ gives

$$\boldsymbol{u} = -\mathbf{B}_{acc}^{-1}(H(\phi)\dot{\boldsymbol{r}} + \mathbf{K_d}(\nabla\phi + \dot{\boldsymbol{r}}) + \mathbf{A}_{acc}\boldsymbol{x}). \tag{3.35}$$

3.4 APF Construction

The form of APF used herein is typical of several other works on APFM spacecraft control [4, 9, 15]. The attractive potential is a quadratic:

$$\phi_{\text{att}} = \frac{1}{2}\boldsymbol{r}^{\intercal}\mathbf{M}\boldsymbol{r}, \tag{3.36}$$

where $\mathbf{M}$ is a positive definite matrix. The repulsive potentials are Gaussian functions:

$$\phi_{\text{rep},i} = \frac{1}{2}\lambda_i \exp(-\tilde{\boldsymbol{r}}_{\text{obs},i}^{\intercal}\mathbf{N}_i\tilde{\boldsymbol{r}}_{\text{obs},i}), \tag{3.37}$$

where the vector $\tilde{\boldsymbol{r}}_{\text{obs},i}$ is the vector pointing from the i^{th} obstacle center to the current position, i.e. $\tilde{\boldsymbol{r}}_{\text{obs},i} = \boldsymbol{r} - \boldsymbol{r}_{\text{obs},i}$. The parameter λ_i determines the overall height of the repulsive potential. The matrix $\mathbf{N}_i$ is a positive definite matrix whose entries determine the oblateness and orientation of the Gaussian function. Because all obstacles considered here are circular, $\mathbf{N}_i$ is diagonal, with the diagonal entries being identical. The magnitude of those diagonal entries determines how sharply the repulsive potential decays with distance from the obstacle center. Values that are too high can cause the vehicle to maneuver away from the obstacle only when it is extremely close, thereby resulting in excessive acceleration. Values that are too low cause the obstacle's influence to extend too far.

Once the values of $\mathbf{M}$ and $\mathbf{N}_i$'s are chosen, the λ_i values are then computed to guarantee collision avoidance. In the cases presented in Sections 3.5 and 4.2, the VAPF Method guarantees that $\dot{\phi}(t) \leq 0$. Therefore, a sufficient (but not necessary) criterion for collision avoidance is that

$$\phi|_{\text{obstacle boundaries}} \geq \phi_0. \tag{3.38}$$

The value of each λ_i is chosen to enforce this criterion. This is accomplished by first calculating the point on each obstacle boundary that is closest to the goal (the point of lowest potential). Then, λ_i is chosen such that the value of $\phi_{\text{att}} + \phi_{\text{rep},i}$ at that location is equal to $\phi_{\text{att}}(\boldsymbol{r}_0)$. This method of calculating λ_i assumes that: 1) the effect of the repulsive potential is negligible at the chase vehicle's initial position, and 2) that the obstacles are far enough apart such that their influence at each other's location is negligible. These assumptions are valid for the geometries studied herein.

3.5 VAPF Method Example Application

The following example demonstrates the application of the VAPF Method in a scenario more complicated than the one-dimensional double integrator. This scenario presents two spacecraft in close proximity undergoing rendezvous in the same orbital

plane. The governing dynamics are the Clohessy-Wiltshire (CW) equations, which are the linearized equations of motion for a chase vehicle (CV) in close proximity to a target vehicle (TV) in circular orbit [18, 19]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 2\omega \\ 0 & 0 & -2\omega & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{3.39}$$

where ω is the magnitude of orbital angular velocity. The TV is at the origin of the reference frame, and the CV, whose position vector is $\boldsymbol{r} = [x \;\; y]^\intercal$, approaches the TV. The attractive potential utilized is a quadratic function, as given in (3.36). Five obstacles are present in the workspace, and their repulsive potentials are Gaussian functions as given in (3.37). The complete equations for ϕ and its first two spatial derivatives are shown below:

$$\phi = \frac{1}{2}\boldsymbol{r}^\intercal \mathbf{M} \boldsymbol{r} + \frac{1}{2}\sum_{i=1}^{5} \lambda_i \exp(-\tilde{\boldsymbol{r}}_{\text{obs},i}^\intercal \mathbf{N}_i \tilde{\boldsymbol{r}}_{\text{obs},i}) \tag{3.40}$$

$$\nabla\phi = \mathbf{M}\boldsymbol{r} - \sum_{i=1}^{5} \lambda_i \exp(-\tilde{\boldsymbol{r}}_{\text{obs},i}^\intercal \mathbf{N}_i \tilde{\boldsymbol{r}}_{\text{obs},i}) \mathbf{N}_i \tilde{\boldsymbol{r}}_{\text{obs},i} \tag{3.41}$$

$$H(\phi) = \mathbf{M} + \sum_{i=1}^{5} \lambda_i \exp(-\tilde{\boldsymbol{r}}_{\text{obs},i}^\intercal \mathbf{N}_i \tilde{\boldsymbol{r}}_{\text{obs},i}) (2\mathbf{N}_i \tilde{\boldsymbol{r}}_{\text{obs},i} \tilde{\boldsymbol{r}}_{\text{obs},i}^\intercal \mathbf{N}_i - \mathbf{N}_i). \tag{3.42}$$

The values of $\mathbf{M}$ and $\mathbf{N}_i$ chosen for this particular scenario are

$$\mathbf{M} = \begin{bmatrix} 0.005 & 0 \\ 0 & 0.005 \end{bmatrix} \tag{3.43}$$

$$\mathbf{N}_i = \begin{bmatrix} 0.0008 & 0 \\ 0 & 0.0008 \end{bmatrix} \qquad \text{for } i = 1 \text{ to } 5. \tag{3.44}$$

The λ_i's were chosen according to the procedure in Section 3.4. Obstacle radii and locations of the obstacle centers are given in Table 3.1. The VAPF Method control

Table 3.1: Obstacle Data for Rendezvous Simulation

	Obstacle 1	Obstacle 2	Obstacle 3	Obstacle 4	Obstacle 5
x_{obs} (m)	100	270	300	-200	160
y_{obs} (m)	200	70	300	300	-100
R (m)	25	25	50	50	25

law in (3.35) can now be used to compute the appropriate control. Note that, in this case, the matrices $\mathbf{A}_{acc}$ and $\mathbf{B}_{acc}$ are

$$\mathbf{A}_{acc} = \begin{bmatrix} 3\omega^2 & 0 & 0 & 2\omega \\ 0 & 0 & -2\omega & 0 \end{bmatrix} \tag{3.45}$$

$$\mathbf{B}_{acc} = \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \end{bmatrix}. \tag{3.46}$$

A simulation using the VAPF Method control law was conducted with the initial conditions,

$$\boldsymbol{x}_0 = \begin{bmatrix} 400 \\ 500 \\ -0.7 \\ -0.3 \end{bmatrix} \begin{matrix} \text{[m]} \\ \text{[m]} \\ \text{[m/s]} \\ \text{[m/s]} \end{matrix}, \tag{3.47}$$

and an orbital angular velocity of $\omega = 0.001127$ rad/s (comparable to that of the International Space Station). Additional parameters are $m = 5$kg and $\mathbf{K}_d = 0.1\mathbf{I}$.

The resulting trajectory is plotted on the surface of ϕ in Figure 3.1. The initial position is marked with a green dot, and a red dot marks the terminal position. The two-dimensional trajectory, state histories, control histories, and VAPF history are shown in Figures 3.2(a) to 3.2(d). The trajectories clearly show that the CV successfully maneuvered around obstacles to avoid collisions and converged to the TV's location (the origin). The VAPF history in Figure 3.2(d), shows that the numerical

results for ϕ_v indeed matched the expected theoretical decay, which was in this case

$$\phi_v(t) = \phi_{v,0} \exp(-0.2t). \tag{3.48}$$

This exponential decay confirms that the CV velocity quickly converged from its initial value to $-\nabla\phi$.

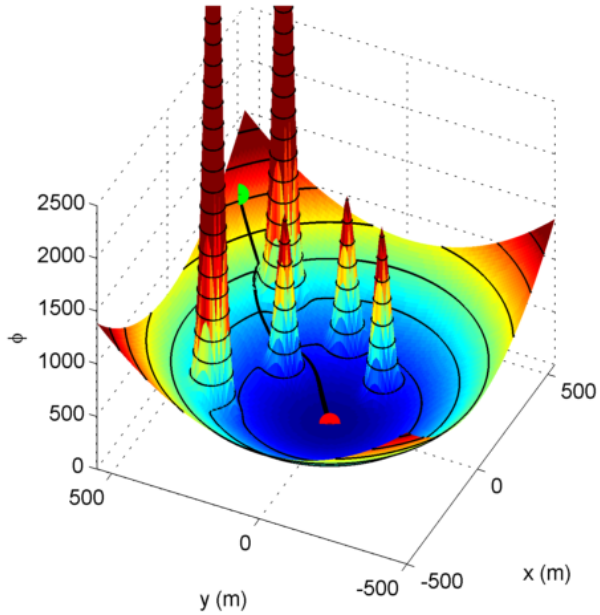
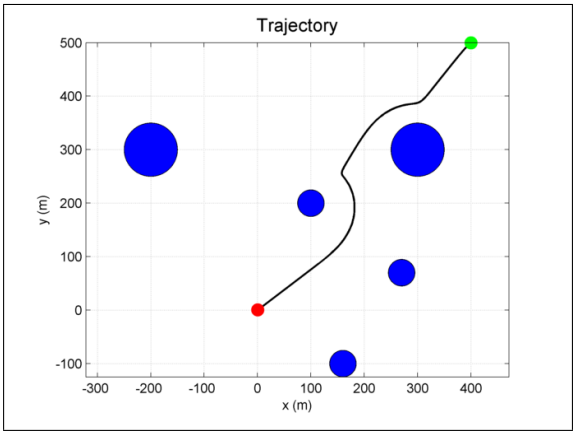
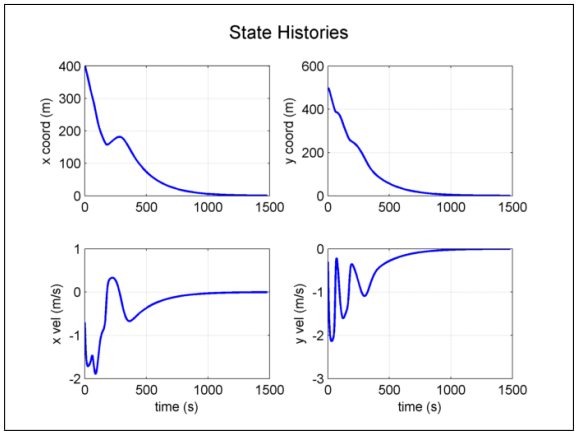


Figure 3.1: Plot of ϕ with Trajectory

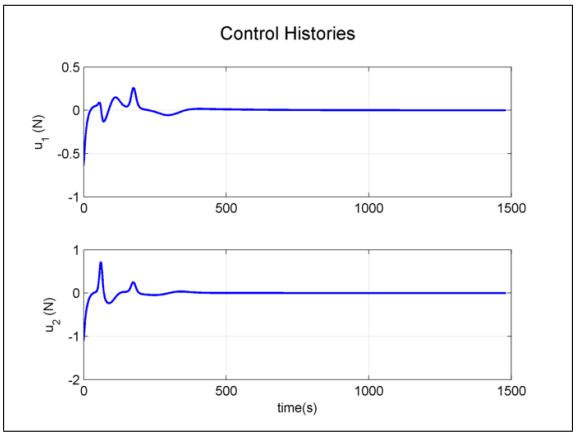
The simulation results also illustrate some of the drawbacks of APFMs that were discussed in Chapter II. First, slow converge in the vicinity of the goal can be noted in Figure 3.2(b). Examination of the trajectory also shows that the CV gave much wider berth to the obstacles than was necessary. The maneuvers to avoid the obstacles were also quite sudden, as evidenced by the peaky control histories in Figure 3.2(c). Of course, these behaviors are simply a result of the CV traveling with velocity $-\nabla\phi$. The VAPF Method, like other APFMs, also has no consideration of a cost function, and therefore the results seen here are suboptimal with respect to typical cost functions, such as control usage. The following chapter investigates



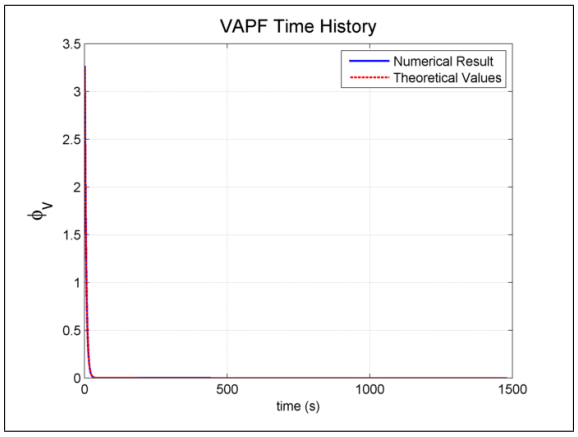
(a) Chase vehicle trajectory



(b) Chase vehicle state histories



(c) Chase vehicle control histories



(d) Velocity APF time history

Figure 3.2: Results for the VAPF Method Example

how to bring some consideration of a cost function into an APFM and achieve more optimal results.

IV. Increasing the Optimality of APFMs

This chapter presents an investigation into increasing the optimality of APFMs. First, optimal control theory is briefly reviewed in Section 4.1. Then, Section 4.2 develops a modification of the VAPF Methods that adds flexibility, thereby providing freedom for optimization. This modified method is put into an optimal control framework in Section 4.3, and four cases of study are also introduced. In Section 4.4, the results for those four cases are presented. The solutions to the original optimal control problem (OCP) and the so-called hybrid OCP are compared in terms of computation time and optimality with respect to the given cost function. Section 4.5 reviews the AAPF developed by Muñoz [4] as another means to increase optimality, and the AAPF is implemented via the VAPF Method.

4.1 Review of Optimal Control

Optimal control has its roots in the calculus of variations developed in the 1600s. In 1639, Galileo Galilei posed one of the earliest OCPs, the brachistochrone problem [20], whose goal is to find the shape of a piece of wire connecting two points such that a bead sliding on the wire travels from the first point to the second in minimum time. Newton, l'Hopital, Leibniz, the Bernoullis, and later Euler and Lagrange were active in the 17^{th} and 18^{th} centuries in applying the calculus of variations to OCPs. In fact, the necessary conditions of optimality for an OCP without path constraints are termed the Euler-Lagrange equations. Since those early days, much progress has been made in optimal control theory, including the development of Pontryagin's Minimum Principle, which gives the necessary conditions for an OCP with control or state constraints [21]. Great advances in numerical solutions to OCPs mean that problems once thought intractable are readily solved today on personal computers [22–24]. This section provides only a brief overview of the vast field of optimal control.

A general optimal control problem for minimizing a scalar cost function J can be stated as

$$\begin{aligned}
&\min_{\boldsymbol{u}} && J = \Phi\left(\boldsymbol{x}_f\right) + \int_{t_0}^{t_f} L\left(\boldsymbol{x}, \boldsymbol{u}, t\right)\, \mathrm{d}t \\
&\text{subject to:} && \\
&\text{the dynamics} && \dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \\
&\text{initial conditions} && t_0 \text{ and } \boldsymbol{x}_0 \\
&\text{final conditions} && \psi\left(\boldsymbol{x}(t_f), t_f\right) = 0 \\
&\text{path constraints} && C\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) \leq 0.
\end{aligned}$$

The solution methodologies for OCPs can be separated into two categories: indirect and direct. Indirect methods use the techniques of the calculus of variations to derive the necessary conditions for optimality. For problems without path constraints these conditions are the well-known Euler-Lagrange equations. For problems with control or state constraints, the necessary conditions are given by Pontryagin's Minimum Principle. Problems with mixed state and control constraints must use even more general necessary condtions [25]. The necessary conditions for optimality result in a boundary value problem (BVP) with coupled differential and algebraic equations. For some simple problems, such as the brachistochrone and sphere geodesic problems, the BVPs can be solved analytically [26]. In general, though, the BVPs must be solved numerically using indirect shooting methods. The essence of indirect shooting methods is to guess values for unknown boundary values of the BVP and then integrate numerically to the other end. The guess for the unknown boundary values is iterated until the terminal conditions at the other end match the known values there. One significant challenge with indirect shooting methods is that they may not converge if the initial guess is too far from the true value [27]. Additionally, state and control constraints further complicate indirect shooting methods because consideration must be given to arcs in the problem where those constraints become active.

Direct solution methods parameterize the controls and/or states and then translate the OCP into a nonlinear programming (NLP) problem [28]. In the so-called direct shooting method, only the control is parameterized into a functional form with unknown parameters. Those parameters then become the independent variables in a NLP problem. For each guessed control sequence, the state dynamics are integrated numerically, and the cost and constraint values calculated. The NLP solver perturbs the parameters for the control until a minimum cost is reached and any path constraints and boundary conditions are satisfied. Collocation methods are another class of direct solution methods, which have been used extensively in recent years. Collocation methods parameterize both the state and control in terms of certain basis functions, such as Lagrange polynomials [24]. The problem is broken up into a number of points in the domain (called collocation points), and the approximating functions are chosen such that the dynamics of the problem are satisfied at those collocation points. An NLP is combined with this collocation scheme to minimize the given cost function. Two popular optimal control software packages, DIDO and GPOPS-II, both utilize collocation methods (in particular, global orthogonal collocation) [29,30]. GPOPS-II is used to obtain the results in Section 4.4.

4.2 Variable Speed and Direction VAPF Method

The VAPF Method developed in Chapter III causes a system's velocity to converge to $-\nabla\phi$. However, the resulting gradient-following trajectories are generally suboptimal because they are based on only the shape of ϕ. In order to find more optimal trajectories, flexibility must be added to allow the system's velocity to deviate from $-\nabla\phi$. An important issue then is how to add this flexibility to the method while still preventing obstacle collisions. The key realization is that when the system is moving, it will avoid obstacle collisions so long as it moves in a direction of nonincreasing ϕ. Stated another way, the vehicle may possess any velocity so long as

$$\dot{\boldsymbol{r}} \cdot \nabla\phi \leq 0. \tag{4.1}$$

In this new method, termed the Variable Direction and Speed VAPF (VDSVAPF) Method, the desired velocity is written not as $-\nabla\phi$, but rather a rotated and scaled vector obtained from $-\nabla\phi$. To ensure nonincreasing ϕ (and therefore collision avoidance), the angle of rotation away from $-\nabla\phi$, θ, is limited to

$$-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}. \tag{4.2}$$

The scalar multiplier, K_S, is limited to nonnegative values. Thus, the desired velocity in the VDSVAPF Method is written as

$$\dot{\boldsymbol{r}} = -K_S \mathbf{K}_R \nabla\phi, \tag{4.3}$$

where $\mathbf{K}_R$ is a rotation matrix defined as

$$\mathbf{K}_R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{4.4}$$

for problems in two dimensions. (Alternate formulations for the desired velocity are discussed in Appendix A.) In a sense, K_S can be thought of as a "gas pedal" which modulates speed (relative to the magnitude of $\nabla\phi$), and $\mathbf{K}_R$ can be thought of as a "steering wheel" which rotates the velocity vector. Another interpretation is to envision the surface of ϕ as a fixed landscape. The system may move over this landscape in any manner so long as ϕ is nonincreasing.

With the new expression for desired velocity now defined, the analysis to determine an acceleration law will now proceed in the same manner as for the VAPF Method. First, the VAPF is defined as

$$\begin{aligned} \phi_v &= \frac{1}{2} \left(\dot{\boldsymbol{r}}_{\text{desired}} - \dot{\boldsymbol{r}}\right)^\intercal \left(\dot{\boldsymbol{r}}_{\text{desired}} - \dot{\boldsymbol{r}}\right) \\ &= \frac{1}{2} \left(K_S \mathbf{K}_R \nabla\phi + \dot{\boldsymbol{r}}\right)^\intercal \left(K_S \mathbf{K}_R \nabla\phi + \dot{\boldsymbol{r}}\right). \end{aligned} \tag{4.5}$$

Now the time derivative of ϕ_v is

$$\dot{\phi}_v = (K_S \mathbf{K}_R \nabla\phi + \dot{\boldsymbol{r}})^\intercal \left(\dot{K}_S \mathbf{K}_R \nabla\phi + K_S \dot{\mathbf{K}}_R \nabla\phi + K_S \mathbf{K}_R H(\phi) \dot{\boldsymbol{r}} + \ddot{\boldsymbol{r}} \right), \tag{4.6}$$

where $\dot{\mathbf{K}}_R$ is

$$\dot{\mathbf{K}}_R = \dot{\theta} \begin{bmatrix} -\sin\theta & -\cos\theta \\ \cos\theta & -\sin\theta \end{bmatrix} \tag{4.7}$$

In order to ensure that $\dot{\boldsymbol{r}}$ converges to $-K_S \mathbf{K}_R \nabla\phi$, an acceleration must be chosen which causes $\dot{\phi}_v \leq 0$. The acceleration is chosen to be

$$\ddot{\boldsymbol{r}} = -\dot{K}_S \mathbf{K}_R \nabla\phi - K_S \dot{\mathbf{K}}_R \nabla\phi - K_S \mathbf{K}_R H(\phi) \dot{\boldsymbol{r}} - \mathbf{K}_D \left(K_S \mathbf{K}_R \nabla\phi + \dot{\boldsymbol{r}} \right) \equiv \ddot{\boldsymbol{r}}_{\text{VDSVAPF}}, \tag{4.8}$$

where again $\mathbf{K}_D$ is a positive definite diagonal matrix. Substitution of (4.8) into (4.6) yields the following expression for $\dot{\phi}_v$

$$\dot{\phi}_v = -\left(K_S \mathbf{K}_R \nabla\phi + \dot{\boldsymbol{r}} \right)^\intercal \mathbf{K}_D \left(K_S \mathbf{K}_R \nabla\phi + \dot{\boldsymbol{r}} \right), \tag{4.9}$$

which is a nonpositive quantity, and therefore $\dot{\boldsymbol{r}}$ converges to $-K_S \mathbf{K}_R \nabla\phi$. The convergence is exponential, as shown in Section 3.3.2.

The VDSVAPF Method has been developed as a way to allow the system velocity to deviate from $-\nabla\phi$ by means of the rotation angle, θ, and the speed scalar, K_S. This added flexibility can be utilized to allow an optimal control solver to find a more optimal trajectory compared to strictly following $-\nabla\phi$. Section 4.3 details how the VDSVAPF Method is placed into an optimal control framework.

4.3 Formulation of the Hybrid Optimal Control Problem

This section first presents the OCP under investigation. From here forward, this problem will be referred to as the "original OCP." Then, the original OCP will be transformed into the so-called "hybrid OCP" by the insertion of the VDSVAPF Method dynamics. Finally, four cases of study are introduced.

4.3.1 The Original OCP and the Hybrid OCP. The OCP of interest in this section (i.e. the original OCP) is a quadratic cost, minimum control satellite rendezvous problem. The problem is given as

$$\begin{aligned}
\min_{\boldsymbol{u}} \quad & J = \int_0^{t_f} \boldsymbol{u}^\intercal \boldsymbol{u} \, \mathrm{d}t \\
\text{subject to} \quad & \text{2-D CW dynamics} \\
& R_j - \|\boldsymbol{r}(t) - \boldsymbol{o}_j\| \leq 0 \qquad \text{for } j = 1 \text{ to } l, \\
& \|\boldsymbol{r}(t_f)\| \leq 1\,\mathrm{m}, \quad \|\dot{\boldsymbol{r}}(t_f)\| \leq 0.1\,\mathrm{m/s} \\
& \boldsymbol{x}(0) = \boldsymbol{x}_0.
\end{aligned}$$

The two-dimensional CW equations are given in (3.39).

Now the hybrid OCP is formed by inserting the VDSVAPF Method into the framework of the original OCP. The first challenge at hand is how to write the original cost integrand ($\boldsymbol{u}^\intercal \boldsymbol{u}$) in terms of the VDSVAPF Method dynamics. For second-order, fully actuated linear dynamics (e.g. the CW dynamics), the system input may be written as

$$\boldsymbol{u} = \mathbf{B}_{acc}^{-1} \left(\ddot{\boldsymbol{r}}_{\text{VDSVAPF}} - \mathbf{A}_{acc} \boldsymbol{x} \right), \tag{4.10}$$

where $\ddot{\boldsymbol{r}}_{\text{VDSVAPF}}$ is given in (4.8). Therefore the cost integrand in the hybrid OCP is

$$L_{\text{VDSVAPF}} = \left(\ddot{\boldsymbol{r}}_{\text{VDSVAPF}} - \mathbf{A}_{acc} \boldsymbol{x} \right)^\intercal \mathbf{B}_{acc}^{-1\intercal} \mathbf{B}_{acc}^{-1} \left(\ddot{\boldsymbol{r}}_{\text{VDSVAPF}} - \mathbf{A}_{acc} \boldsymbol{x} \right). \tag{4.11}$$

The dynamics for the hybrid OCP are the VDSVAPF Method dynamics given in (4.8). Additionally, the obstacle avoidance path constraints are removed from the hybrid OCP because the VDSVAPF dynamics already enforce obstacle avoidance.

The complete statement of the hybrid OCP is then

$$
\begin{aligned}
\min_{\dot{K}_S, \dot{\theta}} \quad & J = \int_0^{t_f} L_{\text{VDSVAPF}} \, \mathrm{d}t \\
\text{subject to} \quad & \ddot{\boldsymbol{r}} = \ddot{\boldsymbol{r}}_{\text{VDSVAPF}} \\
& K_S(t) \geq 0, \quad -\frac{\pi}{2} \leq \theta(t) \leq \frac{\pi}{2} \\
& \|\boldsymbol{r}(t_f)\| \leq 1\,\text{m}, \quad \|\dot{\boldsymbol{r}}(t_f)\| \leq 0.1\,\text{m/s} \\
& \boldsymbol{x}(0) = \boldsymbol{x}_0.
\end{aligned}
$$

The hybrid OCP has two additional states (K_S and θ) compared to the original OCP. The independent variables are no longer the components of thrust $\boldsymbol{u}$, but rather the VDSVAPF Method control parameters $\dot{K}_S$ and $\dot{\theta}$.

4.3.2 Description of Four Test Cases. Four cases will be used to compare the solutions of the original OCP and the hybrid OCP. Case 1 has no obstacles. The initial conditions for the CV are $\boldsymbol{r}(0) = [0, 100]$ m and $\dot{\boldsymbol{r}}(0) = [0, 0]$ m/s, and $t_f = 50$ s. The purpose of Case 1 is to provide the simplest geometry possible, and to verify that the hybrid OCP solution is working. Case 2 introduces a circular obstacle with a 15 m radius at $\boldsymbol{r}_{\text{obs}} = [-10, 50]$ m. The initial conditions and t_f remain the same as in Case 1. Case 3 is nearly identical to Case 2, with the only difference being that the obstacle has been moved 5 meters to $\boldsymbol{r}_{\text{obs}} = [-5, 50]$ m. The primary purpose of Case 3 is to demonstrate the sensitivity of the hybrid OCP solution to obstacle position. Finally, Case 4 is another obstacle-free case. Initial conditions are $\boldsymbol{r}(0) = [0, 300]$ m and $\dot{\boldsymbol{r}}(0) = [0, 1]$ m/s, and the final time is free within the bounds 1900 s $\leq t_f \leq$ 2000 s. The purpose of Case 4 is to demonstrate a case where a gradient-following trajectory looks vastly different from the optimal trajectory. The cases are summarized in Table 4.1.

Table 4.1: Summary of Four Test Cases

	$\boldsymbol{r}(0)$ (m)	$\dot{\boldsymbol{r}}(0)$ (m/s)	t_f (s)	$\boldsymbol{r}_{\text{obs}}$ (m)	Obstacle Radius (m)
Case 1	[0, 100]	[0, 0]	50	n/a	n/a
Case 2	[0, 100]	[0, 0]	50	[-10, 50]	15
Case 3	[0, 100]	[0, 0]	50	[-5, 50]	15
Case 4	[0, 300]	[0, 1]	1900 to 2000	n/a	n/a

4.4 Results for the Hybrid OCP

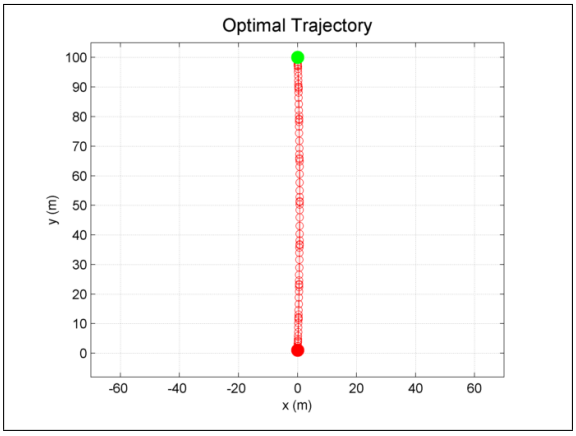
For each of the four cases, both the original OCP and the hybrid OCP were solved using an optimal control solver. A self-written optimal control solver with uniform discretization of the controls was initially implemented, but it required excessive computation times to achieve sufficiently accurate results. Instead, the GPOPS-II optimal control software was used [30]. All solver settings, such as initial mesh guess and required tolerances were identical between the original OCP and the hybrid OCP.

4.4.1 Case 1 Results. The results for Case 1 are shown in Table 4.2 and Figures 4.1 to 4.3. Table 4.2 compares the optimal costs, computation times,

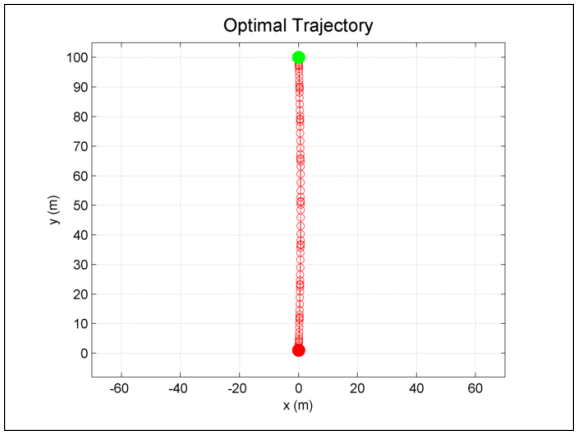
Table 4.2: Tabulated Results for Case 1

	Original OCP	**Hybrid OCP**	**VAPF Method**
Optimal Cost	22.364	22.364	n/a
Computation Time (s)	1.670	3.154	RT
t_f (s)	50.0	50.0	471.1

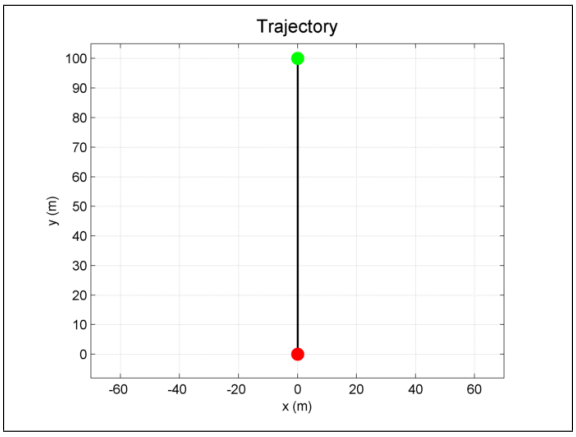
and final time for Case 1. It can be seen that the optimal cost for the hybrid OCP was identical to that of the original OCP. This indicates that the hybrid OCP was set up correctly and produced identical results, despite being a different problem formulation. The cost for the VAPF Method is not listed because comparison of cost would be meaningless due to the fact that a final time cannot be specified for the VAPF Method. The computation time for the original OCP was less than half of that for the hybrid OCP. However, for this simple case, the computation times were on the same order of magnitude. Subsequent cases will show that the hybrid OCP generally



(a) Original OCP

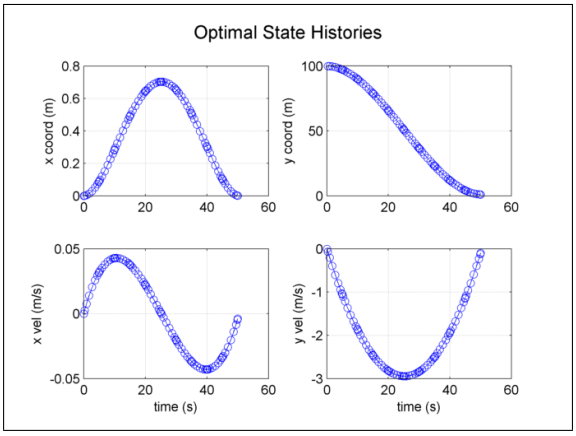


(b) Hybrid OCP

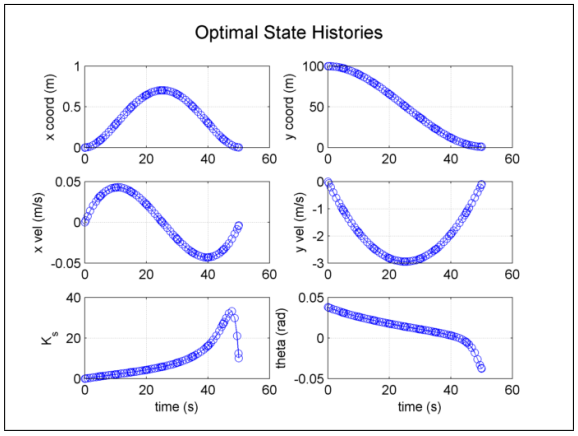


(c) VAPF Method

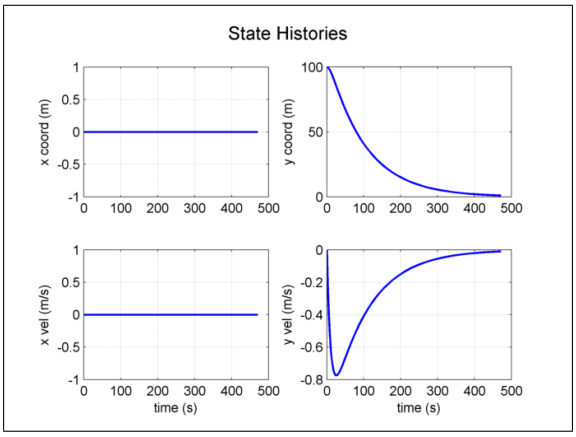
Figure 4.1: Trajectories for Case 1



(a) Original OCP

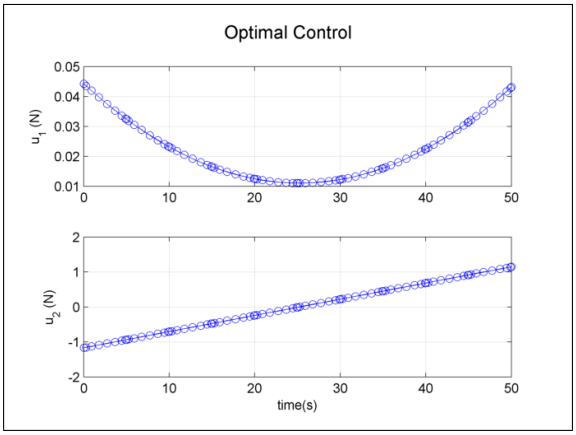


(b) Hybrid OCP

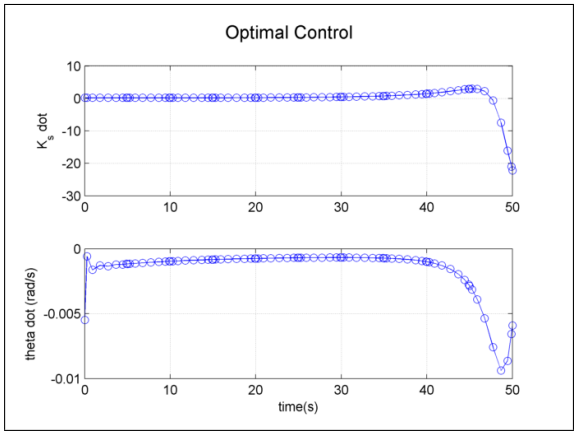


(c) VAPF Method

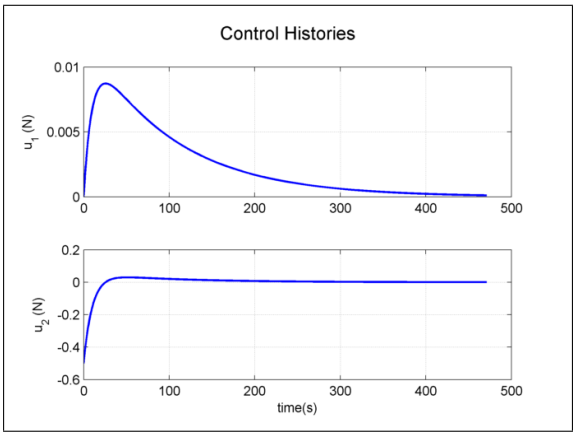
Figure 4.2: State Histories for Case 1



(a) Original OCP



(b) Hybrid OCP



(c) VAPF Method

Figure 4.3: Control Histories for Case 1

requires much more computation time than the original OCP. Computation time for the VAPF Method is listed as "RT" for "real-time" because the VAPF Method requires no pre-computation, and the required control expression can be evaluated instantaneously even with modest computational power.

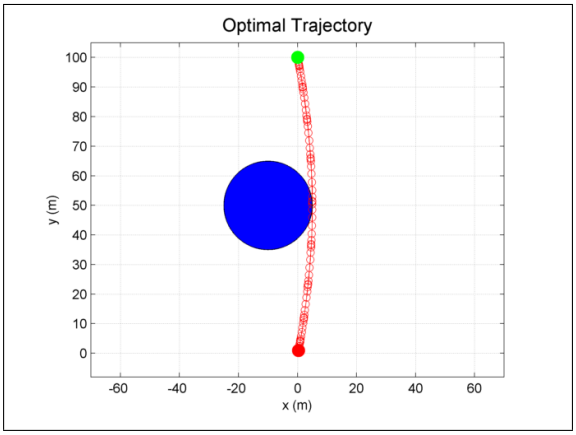
Figure 4.1 shows the trajectories for Case 1. The green and red dots indicate the initial and terminal points, respectively. The trajectories for the original and hybrid OCPs were identical and followed a slightly curved route in order to take advantage of the dynamics of the problem. In contrast, the VAPF Method caused the CV's velocity to converge to $-\nabla\phi$ exactly, creating a direct path toward the goal along the y-axis. The state histories in Figure 4.2 confirm the identical results of the original and hybrid OCPs. The additional states, K_S and θ, are also shown in the state histories for the hybrid OCP. Their varying behavior highlights the need to deviate from $-\nabla\phi$ in order to achieve optimal results. For completeness, the control histories are shown in Figure 4.3.

4.4.2 Case 2 Results. The results for Case 2 are shown in Table 4.3 and Figures 4.4 and 4.5. Case 2 included an obstacle that intruded slightly into the path

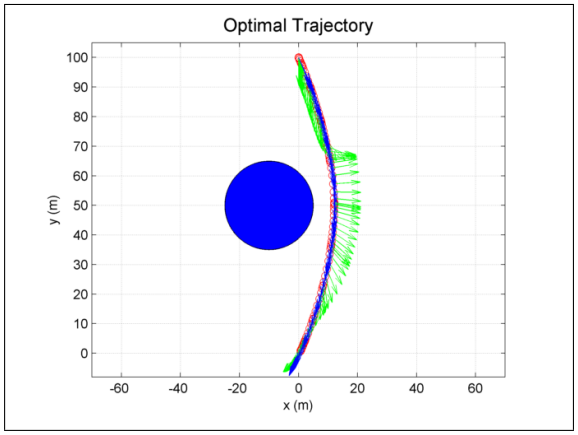
Table 4.3: Tabulated Results for Case 2

	Original OCP	**Hybrid OCP**	**VAPF Method**
Optimal Cost	23.029	27.128	n/a
Computation Time (s)	2.204	22.139	RT
t_f (s)	50.0	50.0	554.2

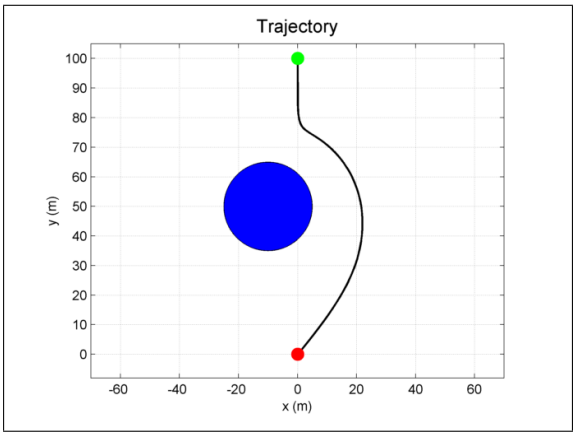
of the original optimal trajectory. From Table 4.3, the most notable result is that the hybrid OCP required a computation time that was an order magnitude larger than that for the original OCP. This result suggests that the solution to the hybrid OCP is more computationally expensive than the solution of the original OCP despite the removal of obstacle-avoidance path constraints from the problem formulation. Another notable feature of Table 4.3 is that the hybrid OCP did not attain as small a cost as the original OCP. An inspection of the trajectories in Figure 4.4 provides



(a) Original OCP

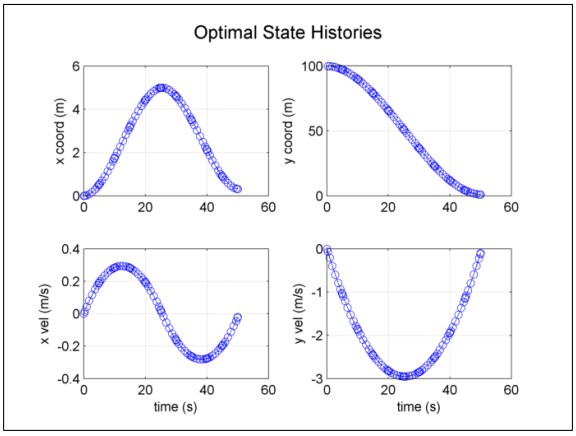


(b) Hybrid OCP

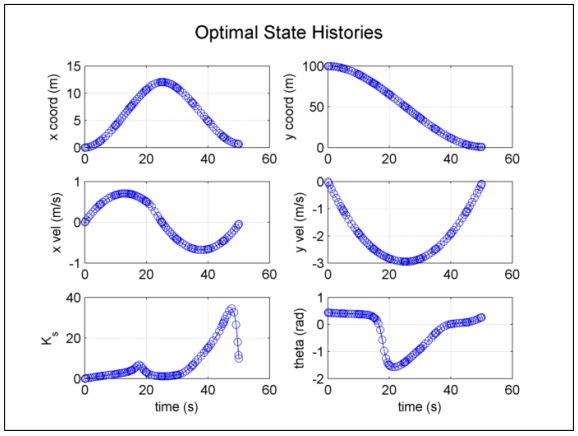


(c) VAPF Method

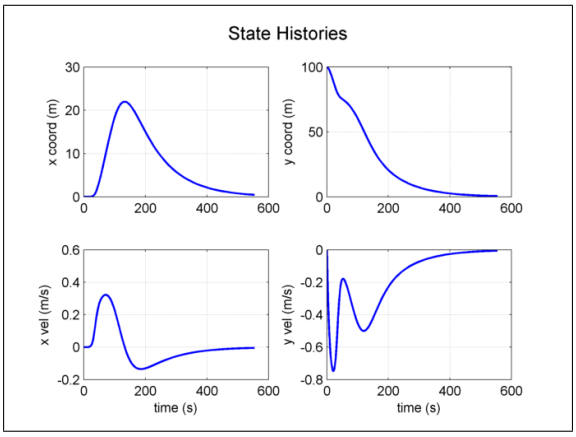
Figure 4.4: Trajectories for Case 2



(a) Original OCP



(b) Hybrid OCP



(c) VAPF Method

Figure 4.5: State Histories for Case 2

an explanation. In the trajectory plot for the hybrid OCP, two sets of arrows are displayed. Blue arrows show the direction of the CV velocity, while green arrows give the direction of $-\nabla\phi$ at each point. (The green and red markers for initial and terminal points have been removed to allow viewing of the arrows.) At the point on the trajectory that is closest to the obstacle, the blue and green arrows are perpendicular. Likewise, the history of θ in Figure 4.5(b) shows that it attained a minimum value of $-\pi/2$ radians. Therefore, the hybrid OCP trajectory brought the CV as close as possible to the obstacle while still maintaining the requirement that $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$ (which enforces $\dot{\phi} \leq 0$). This constraint explains why the hybrid OCP trajectory could not approach the obstacle closer and achieve the same cost as the original OCP.

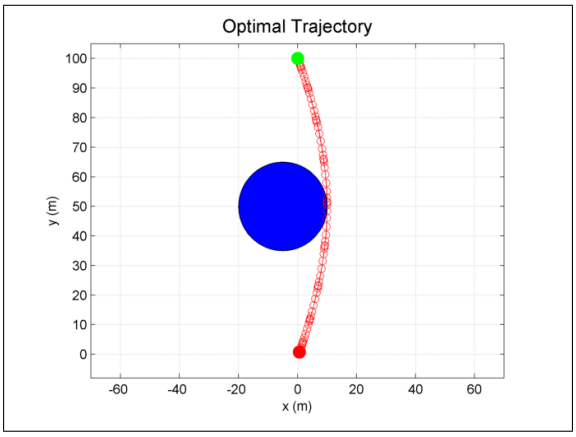
An interesting feature can be observed in the K_S history in Figure 4.5(b). Just before $t = 20$ s, a small peak forms. At this same time, the CV is approaching the obstacle from above. In this region of space, the gradients of ϕ_{att} and ϕ_{rep} point in nearly opposite directions and the magnitude of $-\nabla\phi$ becomes small. Therefore, a peak forms because K_S must increase to keep the CV's speed at the optimal level.

An inspection of the VAPF Method results illustrates some of the advantages and drawbacks of traditional APFMs. The VAPF Method exhibited negligible computational expense. However, it gave the obstacle too wide a berth, and there was no control of the terminal time. The VDSVAPF Method in the hybrid OCP greatly improved upon the quality of the trajectory, but at large computational expense.

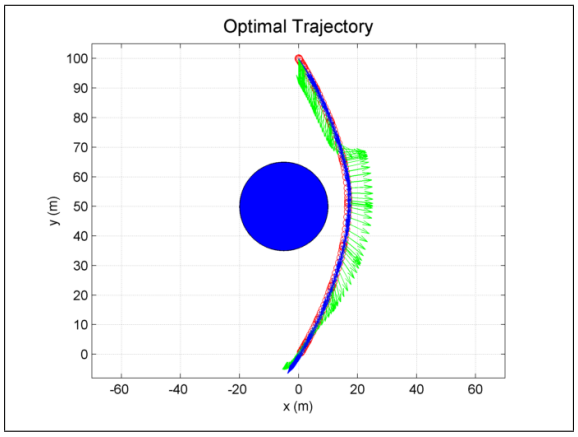
4.4.3 Case 3 Results. Case 3 was nearly identical to Case 2, with the only difference being that the position of the obstacle center was moved 5 m in the $+x$ direction. The primary purpose of Case 3 was to examine the sensitivity of the hybrid OCP solution to obstacle position. The results for Case 3 are shown in Table 4.4 and Figures 4.6 and 4.7. The trajectories and state histories in Figures 4.6 and 4.7 appear very similar to those of Case 2, and the observations made for the Case 2 trajectories and state histories apply to Case 3 as well. However, as shown in Table 4.4, the computation time for the Case 3 hybrid OCP was an order of magnitude larger

Table 4.4: Tabulated Results for Case 3

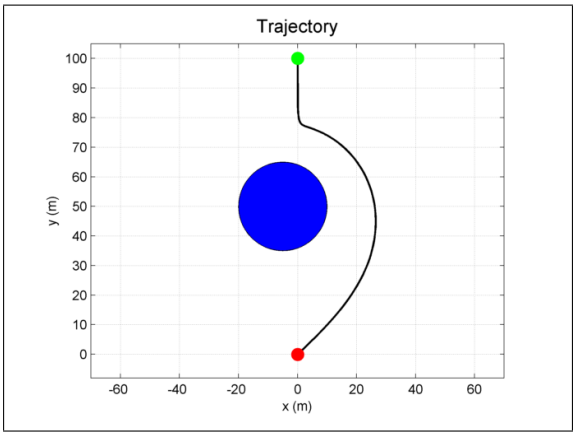
	Original OCP	**Hybrid OCP**	**VAPF Method**
Optimal Cost	25.489	32.003	n/a
Computation Time (s)	2.554	362.907	RT
t_f (s)	50.0	50.0	598.3



(a) Original OCP

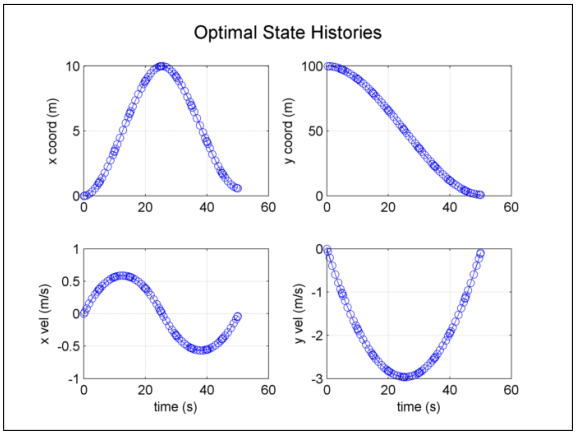


(b) Hybrid OCP

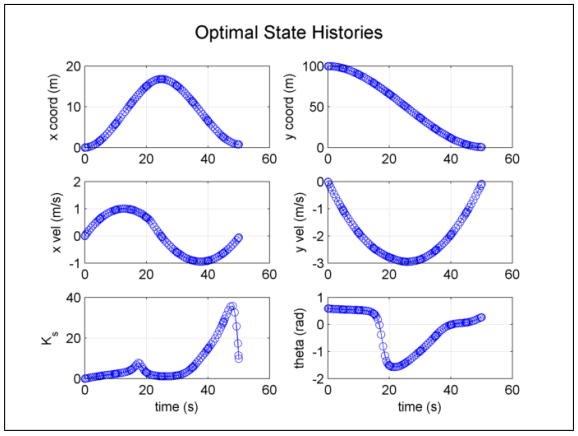


(c) VAPF Method

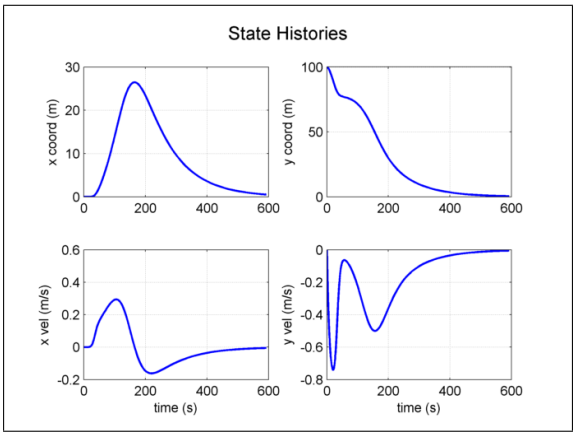
Figure 4.6: Trajectories for Case 3



(a) Original OCP



(b) Hybrid OCP

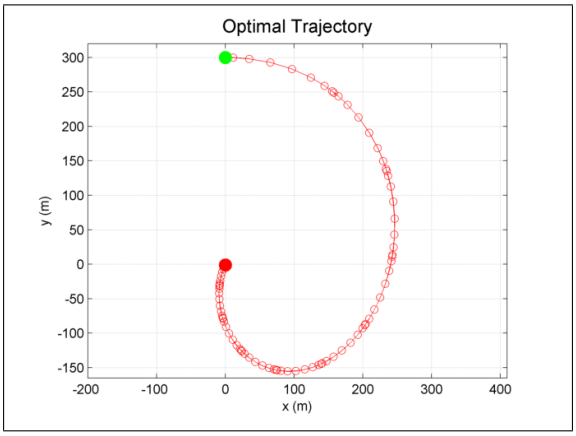


(c) VAPF Method

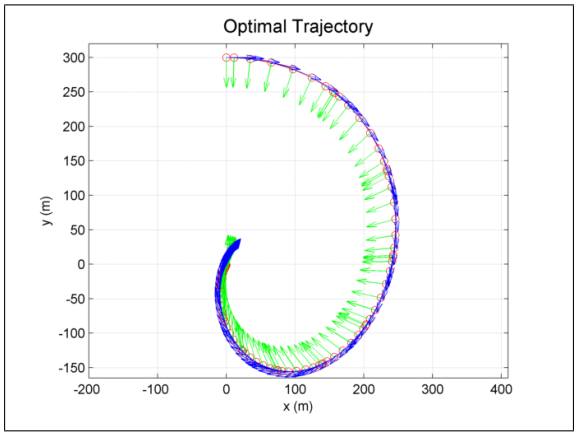
Figure 4.7: State Histories for Case 3

than for Case 2. This demonstrates that the required computation time for hybrid OCP is very sensitive to changes in geometry. Additionally, Case 3 further reinforces the conclusion that the solution of the hybrid OCP is much more computationally expensive than that of the original OCP.

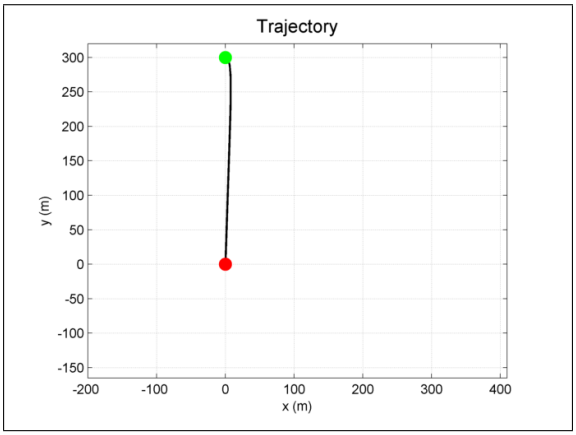
4.4.4 Case 4 Results. Case 4 was another obstacle-free case, but with a free final time between the limits of 1900 s and 2000 s. The trajectory from the original OCP in Figure 4.8(a) shows that the optimal path was a long, looping approach toward the origin. The trajectory from the VAPF Method in Figure 4.8(c), on



(a) Original OCP

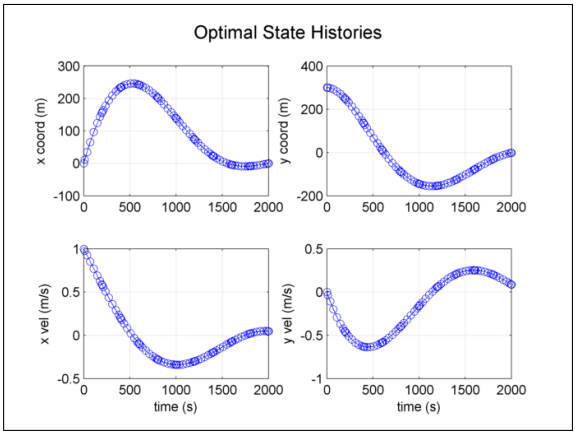


(b) Hybrid OCP

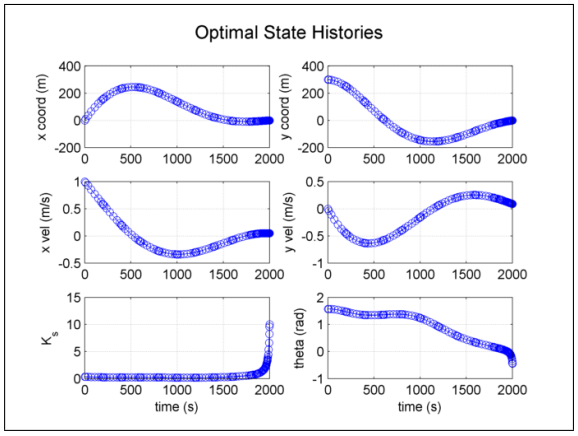


(c) VAPF Method

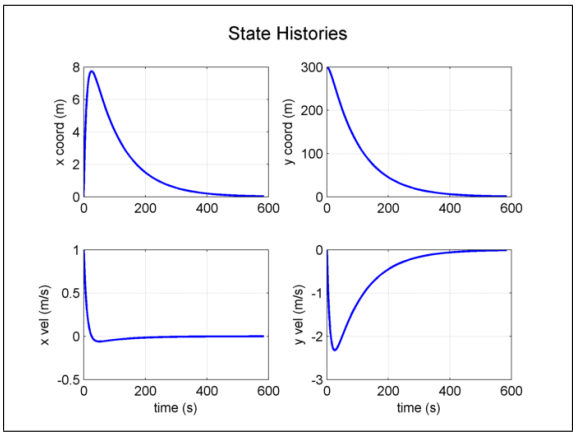
Figure 4.8: Trajectories for Case 4



(a) Original OCP



(b) Hybrid OCP



(c) VAPF Method

Figure 4.9: State Histories for Case 4

the other hand, quickly corrected from the CV's initial velocity and then proceeded straight to the origin with velocity $-\nabla\phi$. The solution from the hybrid OCP was again able to deviate from $-\nabla\phi$ and take advantage of the initial velocity and system dynamics to follow a path that was nearly identical to that of the original OCP, as seen in Figures 4.8(b) and 4.9(b). The results in Table 4.5 show that the hybrid OCP solution attained nearly the same optimal cost as the original OCP. However, the

Table 4.5: Tabulated Results for Case 4

	Original OCP	**Hybrid OCP**	**VAPF Method**
Optimal Cost	0.04217	0.04218	n/a
Computation Time (s)	1.853	114.177	RT
t_f (s)	2000.0	2000.0	584.2

computation time for the hybrid OCP was again much larger than for the original OCP. This high computational expense thus prevents the hybrid OCP from being a useful intermediate method between traditional APFMs and the original OCP.

4.4.5 Discussion of Hybrid OCP Computational Expense. The above results for the hybrid OCP demonstrate that the insertion of an APFM into an optimal control framework was indeed successful in increasing the optimality of APFM results. Optimization was possible because the flexibility in the VDSVAPF Method allowed the CV's velocity to deviate from $-\nabla\phi$ while still avoiding obstacles. There had been some expectation that the solution to the hybrid OCP could be less computationally expensive than the solution to original OCP because obstacle avoidance path constraints were not present in the hybrid OCP. However, these computational savings were not realized, and the hybrid OCP solutions required significantly more computation time than the original OCP.

The large computational expense of the hybrid OCP most likely arose from the higher complexity (in comparison to the original OCP) of the dynamics and of the cost function. In the original OCP, the dynamics are the linear CW equations, given in (3.39). However, the dynamics in the hybrid OCP are given by the acceleration for

the VDSVAPF Method, namely

$$\ddot{\boldsymbol{r}}_{\text{VDSVAPF}} = -\dot{K}_S \mathbf{K}_R \nabla\phi - K_S \dot{\mathbf{K}}_R \nabla\phi - K_S \mathbf{K}_R H(\phi)\dot{\boldsymbol{r}} - \mathbf{K}_D \left(K_S \mathbf{K}_R \nabla\phi + \dot{\boldsymbol{r}}\right).$$

The above dynamics are very nonlinear due to the trigonometric functions (inside $\mathbf{K}_R$), Gaussian functions (inside $\nabla\phi$), and the many products of states and inputs ($\dot{K}_S$ and $\dot{\theta}$). In addition, all of the information about the obstacles is present within the dynamics via the definition of ϕ. In a sense, the complexity due to obstacle avoidance has been removed from path constraints and placed, instead, in the dynamics.

In addition to the dynamics, the cost function integrand became much more complicated for the hybrid OCP. In the original OCP, the cost integrand was simply

$$L = \boldsymbol{u}^\intercal \boldsymbol{u}.$$

In the hybrid OCP, the same quantity is expressed as

$$L = \left(\ddot{\boldsymbol{r}}_{\text{VDSVAPF}} - \mathbf{A}_{acc}\boldsymbol{x}\right)^\intercal \mathbf{B}_{acc}^{-1\intercal} \mathbf{B}_{acc}^{-1} \left(\ddot{\boldsymbol{r}}_{\text{VDSVAPF}} - \mathbf{A}_{acc}\boldsymbol{x}\right), \tag{4.12}$$

a much more complicated expression. The hybrid OCP also includes two more states (K_S and θ) than the original OCP. The nonlinear nature of the dynamics and cost integrand means that the nonlinear programming routine used within GPOPS-II must calculate much more complicated gradients than for the original OCP.

4.5 Adaptive Artificial Potential Function Applied with Continuous Control

Another means of increasing the optimality of APFMs is to use the Adaptive Artificial Potential Function (AAPF) developed by Muñoz [4]. The AAPF is a special type of APF where ϕ_{att} changes shape through time in order to make $-\nabla\phi_{\text{att}}$ match some prescribed velocity profile, $\dot{\boldsymbol{r}}_d$. The prescribed velocity profile, which is chosen by the designer, should impart some degree of optimality with respect to the cost function

being considered. Typically, the choice of $\dot{\boldsymbol{r}}_d$ has no consideration of obstacles. The adapting attractive potential causes the system to travel with prescribed velocity $\dot{\boldsymbol{r}}_d$ when far away from obstacles. If $\dot{\boldsymbol{r}}_d$ happens to drive the system near an obstacle, then the obstacle's repulsive potential (which remains static) "pushes" the system away to avoid a collision. Then, when the system is again far from the obstacle's influence, its velocity converges again to $\dot{\boldsymbol{r}}_d$.

Essentially, the AAPF leverages pre-existing knowledge of obstacle-free velocity profiles which possess some degree of optimality. The system then follows the velocity profile except for when it is perturbed by an obstacle's repulsive potential. One of the attractive features of AAPFs is that there is an analytic expression for the time rate of change of the adapting shaping parameters. Therefore, using an AAPF adds almost no additional computational expense compared to the use of a static APF.

The subsequent section derives the adaptive law for the shaping parameters of ϕ_{att}, with only minor notation changes from the original work of Muñoz. After the AAPF is developed, a continuous control law is needed to cause the system's velocity to match the negative gradient of the time-dependent potential. Because the VAPF Method, as presented in Section 3.3, was derived for time-independent ϕ's, it will be briefly re-derived in Section 4.5.2 to accommodate time-dependent ϕ's. Finally, two example applications are presented in Section 4.5.3.

4.5.1 AAPF Derivation. This section derives the adaptation law for the shaping parameters in an AAPF. The derivation follows Muñoz [4], with the only differences being slight changes in notation and the fact that only the two-dimensional case is presented here. The basic outline of the derivation is as follows: first, define an error, $\boldsymbol{e}$, between the prescribed velocity, $\dot{\boldsymbol{r}}_d$, and $-\nabla\phi$. Then the error time derivative, $\dot{\boldsymbol{e}}$, is written. The time derivatives of the shaping parameters are then chosen in order to make $\dot{\boldsymbol{e}} = -\boldsymbol{e}$, or some positive multiple thereof. This ensures that the components of the error decay exponentially.

The AAPF is defined as a quadratic,

$$\phi = \frac{1}{2} \left(\boldsymbol{r} - \boldsymbol{r}_g \right)^{\intercal} \mathbf{M} \left(\boldsymbol{r} - \boldsymbol{r}_g \right), \tag{4.13}$$

where $\boldsymbol{r}_g$ is the goal location, and $\mathbf{M}$ is a positive semidefinite matrix. To enforce the positive semidefiniteness of $\mathbf{M}$, it is defined in terms of a Cholesky factorization:

$$\mathbf{M} = \mathbf{R}^{\intercal}\mathbf{R}. \tag{4.14}$$

The matrix $\mathbf{R}$ is an upper triangular matrix, termed a Cholesky factor:

$$\mathbf{R} = \begin{bmatrix} \rho_{11} & \rho_{12} \\ 0 & \rho_{22} \end{bmatrix}, \tag{4.15}$$

where each ρ_{ij} is a time-dependent shaping parameter. With this definition, $\mathbf{M}$ is then

$$\mathbf{M} = \begin{bmatrix} \rho_{11}^2 & \rho_{11}\rho_{12} \\ \rho_{11}\rho_{12} & \rho_{12}^2 + \rho_{22}^2 \end{bmatrix}. \tag{4.16}$$

Now, the error is defined as the difference between the prescribed velocity profile, $\dot{\boldsymbol{r}}_d$, and $-\nabla\phi$:

$$\begin{aligned} \boldsymbol{e} &= \dot{\boldsymbol{r}}_d + \nabla\phi \\ &= \dot{\boldsymbol{r}}_d + \mathbf{M}\left(\boldsymbol{r} - \boldsymbol{r}_g \right). \end{aligned} \tag{4.17}$$

Taking the time derivative of (4.17) yields

$$\begin{aligned} \dot{\boldsymbol{e}} &= \ddot{\boldsymbol{r}}_d + \dot{\mathbf{M}}\left(\boldsymbol{r} - \boldsymbol{r}_g \right) + \mathbf{M}\left(\dot{\boldsymbol{r}} - \dot{\boldsymbol{r}}_g \right) \\ &= \ddot{\boldsymbol{r}}_d + \begin{bmatrix} 2\rho_{11}\dot{\rho}_{11} & \left(\rho_{11}\dot{\rho}_{12} + \dot{\rho}_{11}\rho_{12} \right) \\ \left(\rho_{11}\dot{\rho}_{12} + \dot{\rho}_{11}\rho_{12} \right) & 2\left(\rho_{12}\dot{\rho}_{12} + \rho_{22}\dot{\rho}_{22} \right) \end{bmatrix} \left(\boldsymbol{r} - \boldsymbol{r}_g \right) + \mathbf{M}\left(\dot{\boldsymbol{r}} - \dot{\boldsymbol{r}}_g \right). \end{aligned} \tag{4.18}$$

It is desired to choose the $\dot{\rho}_{ij}$'s such that the right-hand side of (4.18) is equal to $-\boldsymbol{e}$ (or some positive multiple thereof). In order to determine what the $\dot{\rho}_{ij}$'s should be, the middle term on the right-hand side of (4.18) must be rearranged with the $\dot{\rho}_{ij}$'s on their own. Multiplying through the middle term and rearranging gives

$$\dot{\boldsymbol{e}} = \ddot{\boldsymbol{r}}_d + \mathbf{S}\dot{\boldsymbol{\rho}} + \mathbf{M}\left(\dot{\boldsymbol{r}} - \dot{\boldsymbol{r}}_g\right), \tag{4.19}$$

where $\dot{\boldsymbol{\rho}} = [\dot{\rho}_{11} \quad \dot{\rho}_{12} \quad \dot{\rho}_{22}]^\intercal$, and $\mathbf{S}$ is

$$\mathbf{S} = \left[\begin{array}{c|c|c} 2\rho_{11}\left(r_1 - r_{1g}\right) + \rho_{12}\left(r_2 - r_{2g}\right) & \rho_{11}\left(r_2 - r_{2g}\right) & 0 \\ \hline \rho_{12}\left(r_1 - r_{1g}\right) & 2\rho_{12}\left(r_2 - r_{2g}\right) + \rho_{11}\left(r_1 - r_{1g}\right) & 2\rho_{22}\left(r_2 - r_{2g}\right) \end{array}\right]. \tag{4.20}$$

In order to solve for $\dot{\boldsymbol{\rho}}$, the right hand side of (4.19) is set equal to $-K_M\boldsymbol{e}$, where K_M is a positive scalar. Solving for $\dot{\boldsymbol{\rho}}$ then yields

$$\dot{\boldsymbol{\rho}} = \mathbf{S}^\intercal\left(\mathbf{S}\mathbf{S}^\intercal\right)^{-1}\left(-\ddot{\boldsymbol{r}}_d - \mathbf{M}\left(\dot{\boldsymbol{r}} - \dot{\boldsymbol{r}}_g\right) - K_M\boldsymbol{e}\right). \tag{4.21}$$

This expression for $\dot{\boldsymbol{\rho}}$ causes $\dot{\boldsymbol{e}} = -K_M\boldsymbol{e}$, and thus the components of the error decay exponentially. It should be noted that the matrix $\mathbf{S}\mathbf{S}^\intercal$ becomes singular when $\boldsymbol{r} = \boldsymbol{r}_g$, so in practice, the adaptation must be halted when the vehicle is in the immediate vicinity of the goal.

4.5.2 Time-dependent APFs in the VAPF Method. Now that the adaptation law for the shaping parameters has been defined in (4.21), a continuous control law is needed which causes the vehicle's velocity to match the negative gradient of the time-dependent AAPF. Previously, the AAPF has been used only in conjunction with an impulsive control law. Because the VAPF Method was originally derived in Section 3.3 for static ϕ's, it will be re-derived here to accommodate time dependence. Again,

the VAPF is defined as

$$\phi_v = \frac{1}{2} \left(\nabla\phi + \dot{\boldsymbol{r}} \right)^{\intercal} \left(\nabla\phi + \dot{\boldsymbol{r}} \right).$$

Now, the time derivative of ϕ_v is

$$\dot{\phi}_v = \left(\nabla\phi + \dot{\boldsymbol{r}} \right)^{\intercal} \left(H\left(\phi\right) \dot{\boldsymbol{r}} + \nabla \frac{\partial \phi}{\partial t} + \ddot{\boldsymbol{r}} \right). \tag{4.22}$$

The acceleration is chosen to make $\dot{\phi}_v$ negative semidefinite:

$$\ddot{\boldsymbol{r}} = -H\left(\phi\right) \dot{\boldsymbol{r}} - \nabla \frac{\partial \phi}{\partial t} - \mathbf{K}_d \left(\nabla\phi + \dot{\boldsymbol{r}} \right), \tag{4.23}$$

where $\mathbf{K}_d$ is again a positive definite, diagonal matrix. In the particular case of the two-dimensional AAPF with stationary goal, the partial time derivative term in (4.23) is given by

$$\nabla \frac{\partial \phi}{\partial t} = \begin{bmatrix} 2\rho_{11}\dot{\rho}_{11} & \left(\rho_{11}\dot{\rho}_{12} + \dot{\rho}_{11}\rho_{12}\right) \\ \left(\rho_{11}\dot{\rho}_{12} + \dot{\rho}_{11}\rho_{12}\right) & 2\left(\rho_{12}\dot{\rho}_{12} + \rho_{22}\dot{\rho}_{22}\right) \end{bmatrix} \left(\boldsymbol{r} - \boldsymbol{r}_g \right), \tag{4.24}$$

and the $\dot{\rho}_{ij}$ values are calculated from (4.21). Substitution of the acceleration from (4.23) into (4.22) yields

$$\dot{\phi}_v = -\left(\nabla\phi + \dot{\boldsymbol{r}} \right)^{\intercal} \mathbf{K}_d \left(\nabla\phi + \dot{\boldsymbol{r}} \right), \tag{4.25}$$

which is a negative semidefinite quantity. Therefore, the acceleration in (4.23) causes $\dot{\boldsymbol{r}}$ to converge exponentially to $-\nabla\phi$, as shown in Section 3.3. For second-order, fully actuated linear systems, the control required to achieve this acceleration is

$$\boldsymbol{u} = \mathbf{B}_{acc}^{-1} \left(\ddot{\boldsymbol{r}} - \mathbf{A}_{acc} \boldsymbol{x} \right), \tag{4.26}$$

.

4.5.3 AAPF Example Applications. The key to achieving some degree of optimality with an AAPF is the selection of the prescribed velocity profile, $\dot{\boldsymbol{r}}_d$. For linear systems, on such choice for $\dot{\boldsymbol{r}}_d$ could be velocity which results from linear quadratic regulator (LQR) control. This section presents idealized satellite rendezvous scenarios which utilize a steady-state LQR velocity profile for $\dot{\boldsymbol{r}}_d$. The cost function used here to generate the steady-state LQR solution is

$$J = \int_0^\infty \boldsymbol{x}^\intercal \mathbf{Q} \boldsymbol{x} + \boldsymbol{u}^\intercal \boldsymbol{u} \, \mathrm{d}t, \tag{4.27}$$

where the positive definite, diagonal matrix $\mathbf{Q}$ was chosen with very small elements to approximate the minimum control stabilizing solution. The resulting optimal control is of the form

$$\boldsymbol{u}^* = \mathbf{K}_{LQR} \boldsymbol{x}, \tag{4.28}$$

where $\mathbf{K}_{LQR}$ is a constant gain matrix [31]. This control law can be substituted back into the linear dynamics to determine the $\dot{\boldsymbol{r}}_d$ and $\ddot{\boldsymbol{r}}_d$ to use in the AAPF formulation:

$$\begin{bmatrix} \dot{\boldsymbol{r}}_d \\ \ddot{\boldsymbol{r}}_d \end{bmatrix} = \left(\mathbf{A} - \mathbf{B}\mathbf{K}_{LQR} \right) \boldsymbol{x}. \tag{4.29}$$

And for a second-order, fully actuated linear system in the same state-space form as (3.34), this gives

$$\dot{\boldsymbol{r}}_d = \dot{\boldsymbol{r}} \tag{4.30}$$

$$\ddot{\boldsymbol{r}}_d = \left(\mathbf{A}_{acc} - \mathbf{B}_{acc}\mathbf{K}_{LQR} \right) \boldsymbol{x}. \tag{4.31}$$

The first cases has no obstacles. The initial conditions for the CV are $\boldsymbol{r}(0) = [0, 300]$ m and $\dot{\boldsymbol{r}}(0) = [1, 0]$ m/s. The scalar K_M is set as 0.1, and the initial value for $\mathbf{M}$ is

$$\mathbf{M}(0) = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}. \tag{4.32}$$

Figure 4.10 is an animation of the AAPF and the CV traveling under its influence. (Please note that animations are guaranteed to appear correctly only in Adobe Acrobat and Adobe Reader software.) The initial value of $\mathbf{M}$ gives a $-\nabla\phi$ which does not

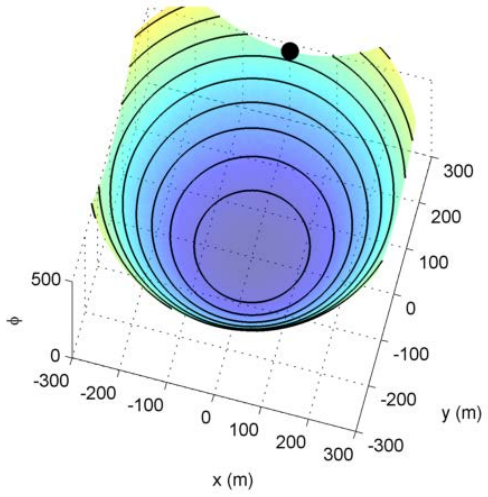
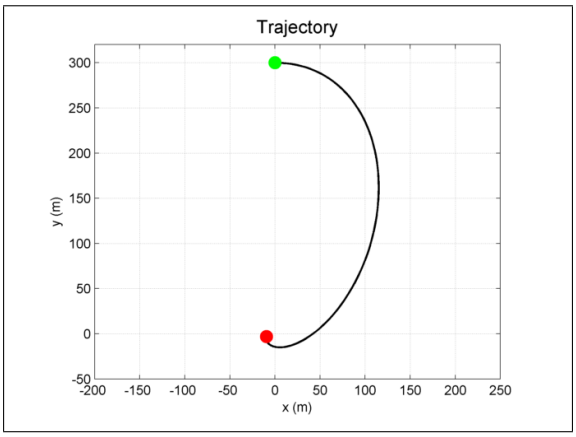


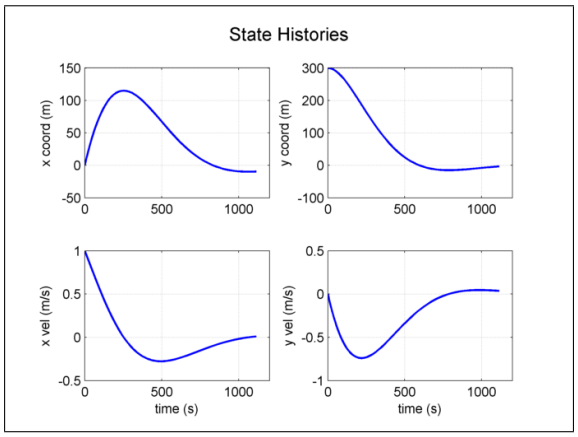
Figure 4.10: Animation of AAPF with Steady-state LQR Prescribed Velocity and No Obstacles (Click image to play)

match $\dot{\boldsymbol{r}}_d$, and the adaptation law quickly flattens and skews ϕ, thereby decreasing the error between $\dot{\boldsymbol{r}}_d$ and $-\nabla\phi$ exponentially. As the animation progresses, ϕ continually adapts to drive the CV along a curved path toward the origin, just as the LQR control would have done. Figure 4.11 displays the two-dimensional trajectory, state histories, control histories, and ρ histories for the simulation.

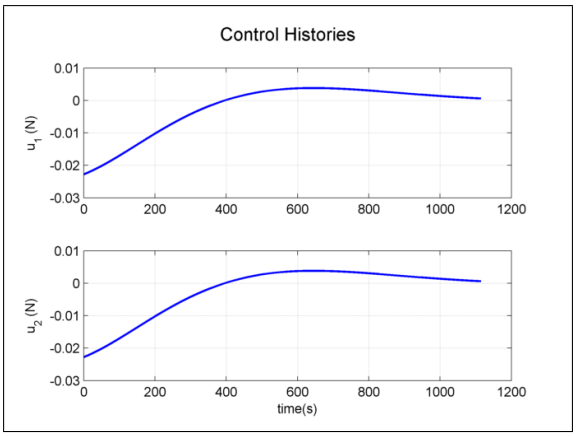
Now two obstacles will be placed into the geometry. One of the obstacles will intersect the original path from the obstacle-free case. An important consideration is the form of the repulsive potentials used for the obstacles in an AAPF. Because the attractive potential continuously adapts, there is a possibility that it could overpower the Gaussian repulsive potentials used previously, thereby allowing a collision. Therefore, repulsive potentials which go to infinity at the obstacle boundaries should



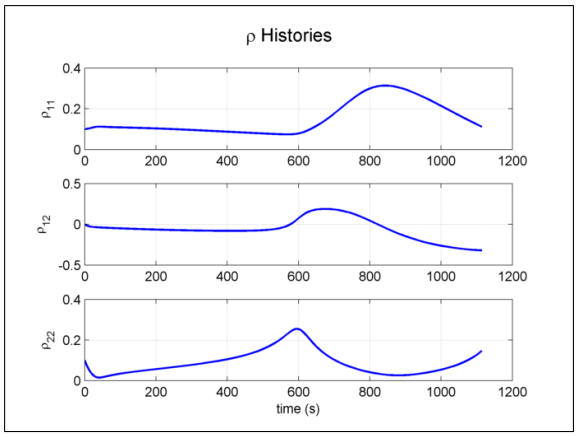
(a) Chase vehicle trajectory



(b) Chase vehicle state histories



(c) Chase vehicle control histories



(d) AAPF ρ histories

Figure 4.11: Results for AAPF Simulation with Steady-state LQR Prescribed Velocity and No Obstacles

be used. The repulsive potentials used here are of the form:

$$\phi_{\text{rep},i} = \frac{\lambda_i}{\left(\exp\left(\tilde{\boldsymbol{r}}_{\text{obs},i}^{\mathsf{T}}\mathbf{N}_i\tilde{\boldsymbol{r}}_{\text{obs},i} - R_i^2\right) - 1\right)^2}, \tag{4.33}$$

where R_i is the radius of the i^{th} obstacle, and the definitions of λ_i, $\tilde{\boldsymbol{r}}_{obs}$ and $\mathbf{N}_i$ are the same as in Section 3.4.

Figure 4.12 is an animation for the case with two obstacles. Until the CV comes

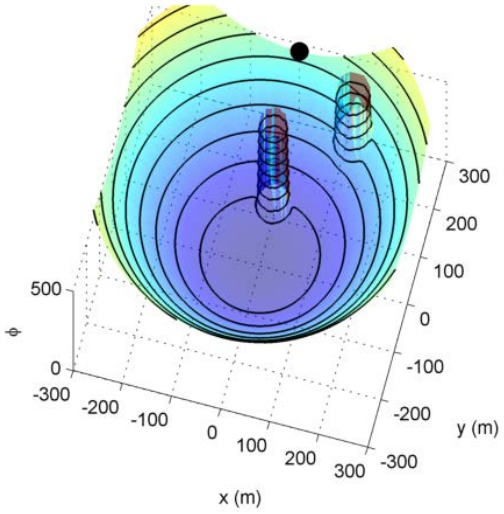
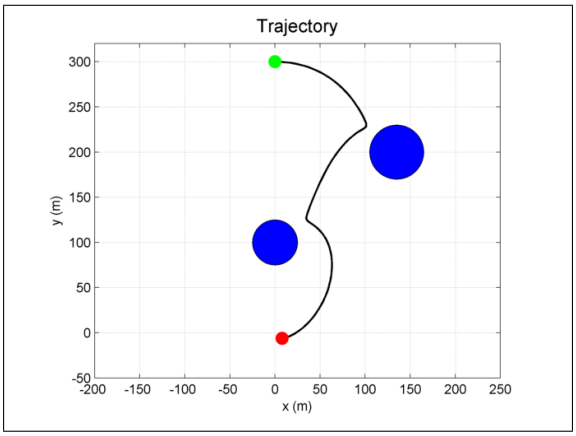


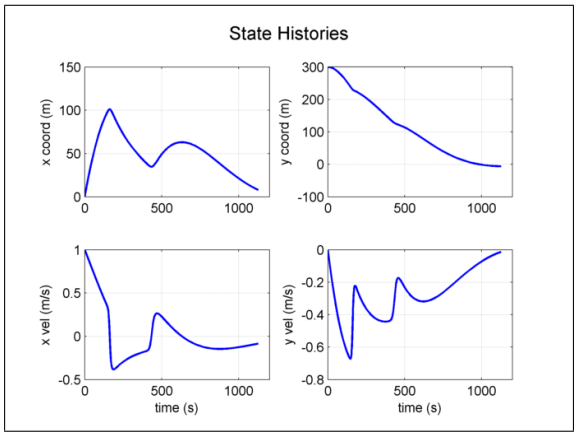
Figure 4.12: Animation of AAPF with Steady-state LQR Prescribed Velocity and Two Obstacles (Click image to play)

near to an obstacle, its trajectory (and the shape of ϕ) appears nearly identical to that in Figure 4.10. When the repulsive obstacle potentials "push" the CV away, the attractive potential adapts to the perturbed velocity and continues to cause $-\nabla\phi_{\text{att}}$ to converge to $\dot{\boldsymbol{r}}_d$. The cost function value for the AAPF simulation here cannot be compared to a numerical solution of the original OCP in a completely fair manner because the AAPF simulation is based on an infinite horizon cost function, and also because the AAPF simulation must be terminated early to avoid $\mathbf{S}\mathbf{S}^{\mathsf{T}}$ becoming nearly singular in the vicinity of the origin. The two-dimensional trajectory, state histories, control histories, and ρ histories are shown in Figure 4.13. The effect of the distur-

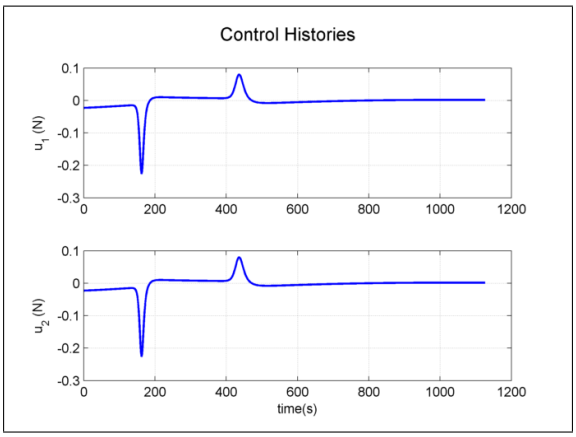
bance from the obstacles may be first seen in the state and control plots near $t = 200$ s.



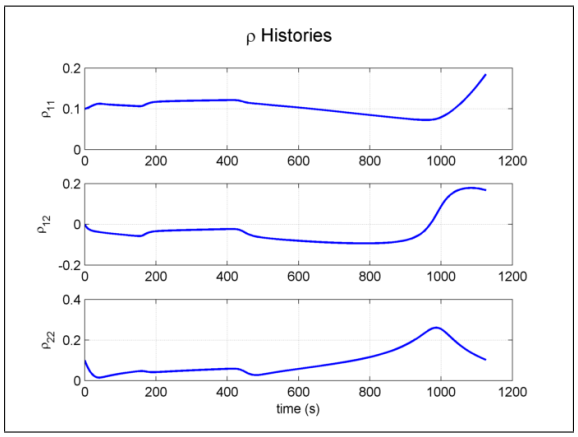
(a) Chase vehicle trajectory



(b) Chase vehicle state histories



(c) Chase vehicle control histories



(d) AAPF ρ histories

Figure 4.13: Results for AAPF Simulation with Steady-state LQR Prescribed Velocity and Two Obstacles

In summary, this section has implemented an AAPF by means of the VAPF Method. Using an AAPF is a computationally efficient way of increasing optimality via the specification of an $\dot{\boldsymbol{r}}_d$ with optimal characteristics. In some cases, such as those with very complicated system dynamics, densely packed obstacles, or complicated cost functions, such an $\dot{\boldsymbol{r}}_d$ may be unknown. But for cases with well-understood dynamics

and sparsely packed obstacles, implementation of an AAPF is an attractive choice due to the negligible increase in computational expense compared to traditional APFMs.

4.6 Summary and Future Avenues for Optimization

This chapter has presented an investigation into increasing the optimality of APFMs. The VAPF Method was augmented with more velocity flexibility to become the VDSVAPF Method. The original OCP was then translated into the hybrid OCP by insertion of the VDSVAPF Method dynamics. In the hybrid OCP, the independent variables were no longer the physical thrust values for the CV, but rather the control parameters $\dot{K}_S$ and $\dot{\theta}$ from the VDSVAPF Method. The hybrid OCP formulation also allowed for the removal of obstacle avoidance path constraints from the problem. Numerical solutions to the original and hybrid OCPs on four test cases showed that, for simple geometries, the hybrid OCP solution matched the original OCP solution exactly. For cases with obstacles, the hybrid OCP solution came as close as possible to the original OCP solution while still respecting the limits on θ. However, the hybrid OCP approached proved to be excessively computationally expensive due to its more complicated dynamics and cost function representation.

Then, the AAPF was implemented with the VAPF Method. The AAPF formulation allows the designer to impart optimality from a foreknown velocity profile into the time-varying shape of the AAPF. In addition, the adaptation law for the shaping parameters of the AAPF is an analytical expression, meaning that use of an AAPF adds negligible computational expense compared to a static APF. The AAPF was implemented via the VAPF Method in two example cases using prescribed velocity profile resulting from steady-state LQR control. This approach proved to be an attractive choice due to its ability to obtain more optimal APFM results with negligible added computational expense.

Apart from the methods addressed in this chapter, other avenues for increasing the optimality of APFMs may be studied in future research. First, a new hybrid OCP may be formulated which uses the APF shaping parameters, rather than APFM

control parameters, as independent variables. While there can be many shaping parameters within a given APF (especially with a large number of obstacles), an appropriate first step might be to focus only on the attractive potential shaping parameters. Based on the observations of the hybrid OCP results in this chapter, however, it is expected that a hybrid OCP using APF shaping parameters would exhibit similarly large computational expense. Another avenue for investigation could be a *static* optimization over APF shaping parameters, i.e. finding the best constant shape for an APF. While this approach would not match the true optimal solution as closely as a hybrid OCP, it would likely be less computationally expensive.

V. Conclusion

Continuous control APFMs and methods of addressing their suboptimality have been studied in this thesis. The Velocity Artificial Potential Function Method developed in Chapter III is the first known continuous control APFM which causes a system's velocity to converge to $-\nabla\phi$. Applicable to general forms of ϕ and any fully actuated system, the VAPF Method defines a second potential function (the velocity artificial potential function) in the velocity space of the vehicle which has $-\nabla\phi$ as its minimum point. An acceleration law is determined which causes the time derivative of the VAPF to be negative semidefinite and gives exponential convergence of the system's velocity toward $-\nabla\phi$. The acceleration law is then inserted into a given system's dynamics to solve for the appropriate control. The method was applied to an idealized spacecraft rendezvous scenario with obstacles. The numerical results confirmed the theoretical convergence propertices, and showed that the chase vehicle was successfully driven to the target vehicle while avoiding obstacles.

The second half of this manuscript investigated methods of addressing the suboptimal results that APFMs typically produce and whether improvements in optimality can be made while preserving the computational efficiency which makes APFMs attractive. The first approach was to insert a continuous control APFM directly into an optimal control framework, forming the so-called hybrid OCP. Because the APFM would eliminate the need for obstacle avoidance path constraints in the problem formulation, some computational savings were thought to be possible in comparison to the original OCP. The VAPF Method was modified into the VDSVAPF Method, which allows the system velocity to deviate from $-\nabla\phi$ while still avoiding obstacles; this modification allowed the freedom needed for optimization. Results from solving the original and hybrid OCPs on four test cases produced several conclusions. First, on simple geometries without obstacles, the hybrid OCP solution was able to exactly match the solution to the original OCP, despite being a different formulation. The flexibility of the VDSVAPF Method allowed several desirable capabilities not easily achievable with traditional APFMs, such as specification of terminal time and more

gradual obstacle avoidance maneuvers. However, in all but the simplest case, the computational time required to solve the hybrid OCP was significantly larger than for the original OCP. This increase in computational expense was most likely due to the greater complexity of the VDSVAPF Method dynamics and the fact that the cost function expression was more complicated in the hybrid OCP. This high computational expense precludes the usefulness of the hybrid OCP solution as an intermediate between traditional APFMs and the original OCP, at least in cases similar to the ones studied in this thesis.

The second approach utilized Muñoz's Adaptive Artificial Potential Function [4], which allows the optimality from foreknown desirable velocity profiles to be imparted into the APF formulation. The AAPF changes shape through time to cause the negative gradient of its attractive potential to match the prescribed velocity profile. One attractive aspect of the AAPF is that the adaptive behavior of the shaping parameters is given through an analytical expression, and thus use of an AAPF adds only negligible computational expense compared to a static APF. The AAPF was implemented in a continuous control scheme for the first time via the VAPF Method. Two example cases using a prescribed velocity resulting from LQR control showed that using an AAPF is a useful way of adding some degree of optimality to APFM results with minimal added computational expense.

There are several areas where future work related to these topics may provide useful results. One of the drawbacks of APFMs that this thesis did not address is that of local minima. Several schemes for addressing the local minimum problem in APFMs have been developed previously, and it would be interesting to see one of those schemes combined with the VAPF Method. Regarding the hybrid OCP, there is room for investigating its computational expense on problems with different types of cost functions and more complicated dynamics; there may exist some problems for which the hybrid OCP is indeed a useful intermediate between traditional APFMs and the orignal OCP. Finally, there are other avenues which could be investigated for achieving optimal results with APFMs. One such avenue would be putting an

APFM into an optimal control framework where time-varying shaping parameters of an APF, rather than the APFM control parameters, are the independent variables. Additionally, performing a static optimization over APF shaping parameters (i.e. finding the best constant shape for an APF) could prove to be a less computationally expensive way to obtain more optimal results from APFMs.

Appendix A. Alternate Formulations for the VDSVAPF Method

As it is presented in Section 4.2, the desired velocity in the VDSVAPF Method is defined as

$$\dot{\boldsymbol{r}} = -K_S \mathbf{K}_R \nabla\phi. \tag{A.1}$$

This formulation necessitates limits on K_S and θ in order to maintain the requirement of nonincreasing ϕ, i.e. $K_S \geq 0$ and $\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$. Because it is generally desirable to have as few constraints as possible in an optimal control problem, two alternate formulations for the desired velocity in the VDSVAPF Method were tested. These alternate formulations removed some of the constraints in the hybrid OCP.

The first alternate formulation defines the desired velocity as

$$\dot{\boldsymbol{r}} = -K_S^2 \mathbf{K}_R \nabla\phi. \tag{A.2}$$

Now that the K_S term is squared, the $K_S \geq 0$ constraint may be removed from the hybrid OCP. With this formulation, the VAPF is now

$$\phi_v = \frac{1}{2}\left(K_S^2 \mathbf{K}_R \nabla\phi + \dot{\boldsymbol{r}}\right)^{\mathsf{T}}\left(K_S^2 \mathbf{K}_R \nabla\phi + \dot{\boldsymbol{r}}\right). \tag{A.3}$$

Taking the time derivative of (A.3) yields

$$\dot{\phi}_v = \left(K_S^2 \mathbf{K}_R \nabla\phi + \dot{\boldsymbol{r}}\right)^{\mathsf{T}}\left(2K_S \dot{K}_S \mathbf{K}_R \nabla\phi + K_S^2 \dot{\mathbf{K}}_R \nabla\phi + K_S^2 \mathbf{K}_R H(\phi)\dot{\boldsymbol{r}} + \ddot{\boldsymbol{r}}\right). \tag{A.4}$$

The acceleration to make $\dot{\phi}_v$ negative semidefinite is now

$$\ddot{\boldsymbol{r}} = -2K_S \dot{K}_S \mathbf{K}_R \nabla\phi - K_S^2 \dot{\mathbf{K}}_R \nabla\phi - K_S^2 \mathbf{K}_R H(\phi)\dot{\boldsymbol{r}} - \mathbf{K}_D\left(K_S^2 \mathbf{K}_R \nabla\phi + \dot{\boldsymbol{r}}\right). \tag{A.5}$$

When this alternate formulation was implemented and solved in GPOPS-II, the computation times were actually marginally larger than those for the original VDSVAPF Method (Section 4.4), despite the removal of the nonnegativity constraint on K_S. This was most likely due to the additional state-input product term, $2K_S\dot{K}_S$,

which is not present in the original formulation. Because of its higher computation times, this alternate formulation was not used.

The second alternate formulation does away with K_S and θ entirely. Recall that the purpose of K_S and θ is to allow the vehicle's velocity to deviate from $-\nabla\phi$ so long as its direction is within $\pi/2$ radians of $-\nabla\phi$. However, this same effect can be achieved by applying a single Cholesky-factored matrix to $-\nabla\phi$. The terms in the Cholesky factors may take any real value, and the resulting matrix is guaranteed to be positive semidefinite. With this in mind, the desired velocity is defined as

$$\dot{\boldsymbol{r}} = -\mathbf{R}^\intercal\mathbf{R}\nabla\phi, \tag{A.6}$$

where the Cholesky factor $\mathbf{R}$ is

$$\mathbf{R} = \begin{bmatrix} \rho_{11} & \rho_{12} \\ 0 & \rho_{22} \end{bmatrix}. \tag{A.7}$$

This formulation has no constraints on the values of the ρ_{ij}'s, and also the trigonometric functions (which were inside the rotation matrix $\mathbf{K}_R$) have been removed. With this definition for desired velocity, the VAPF is now

$$\phi_v = \frac{1}{2}\left(\mathbf{R}^\intercal\mathbf{R}\nabla\phi + \dot{\boldsymbol{r}}\right)^\intercal\left(\mathbf{R}^\intercal\mathbf{R}\nabla\phi + \dot{\boldsymbol{r}}\right). \tag{A.8}$$

Taking the time derivative of ϕ_v gives

$$\dot{\phi}_v = \left(\mathbf{R}^\intercal\mathbf{R}\nabla\phi + \dot{\boldsymbol{r}}\right)^\intercal\left((\dot{\mathbf{R}^\intercal\mathbf{R}})\nabla\phi + \mathbf{R}^\intercal\mathbf{R}H(\phi)\dot{\boldsymbol{r}} + \ddot{\boldsymbol{r}}\right), \tag{A.9}$$

where

$$(\dot{\mathbf{R}^\intercal\mathbf{R}}) = \begin{bmatrix} 2\rho_{11}\dot{\rho}_{11} & (\rho_{11}\dot{\rho}_{12} + \dot{\rho}_{11}\rho_{12}) \\ (\rho_{11}\dot{\rho}_{12} + \dot{\rho}_{11}\rho_{12}) & 2\left(\rho_{12}\dot{\rho}_{12} + \rho_{22}\dot{\rho}_{22}\right) \end{bmatrix}. \tag{A.10}$$

The acceleration to make $\dot{\phi}_v$ negative semidefinite is now

$$\ddot{\boldsymbol{r}} = -(\dot{\mathbf{R}^{\intercal}\mathbf{R}})\nabla\phi - \mathbf{R}^{\intercal}\mathbf{R}H(\phi)\dot{\boldsymbol{r}} - \mathbf{K}_D\left(\mathbf{R}^{\intercal}\mathbf{R}\nabla\phi + \dot{\boldsymbol{r}}\right). \tag{A.11}$$

When this second alternate formulation was implemented and solved in GPOPS-II, the computation times were significantly larger than those for the original VDSVAPF Method (Section 4.4), despite the removal of two constraints (on K_S and θ) and the deletion of trigonometric functions. The suspected reasons for the larger computational expense are twofold. First, this alternate formulation adds one additional state and one additional input compared to the original formulation (there are three ρ_{ij}'s and three $\dot{\rho}_{ij}$'s). Secondly, there are many more state-input product terms in this formulation, as can be seen in (A.10). Because of its larger computation times, this alternate formulation was not used.

References

1. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–99, 1986.

2. I. Lopez and C. R. McInnes, "Autonomous rendezvous using artificial potential function guidance," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 2, pp. 237–241, 1995.

3. M. T. Wolf and J. Burdick, "Artificial potential functions for highway driving with collision avoidance," in *IEEE International Conference on Robotics and Automation*, IEEE, May 2008.

4. J. D. Muñoz, *Rapid Path-planning Algorithms for Autonomous Proximity Operations of Satellites*. PhD thesis, University of Florida, 2011.

5. O. Khatib, *Commande dynamique dans l'espace operational des robots manipulateurs en presence d'obstacles*. PhD thesis, Ecole Nationale Superieure de l'Aeronatique et de l'Espace, 1980.

6. M. Ivanescu, N. Popescu, and D. Popescu, *A Variable Length Tentacle Manipulator Control System*, pp. 3274–3279. Institute of Electrical and Electronics Engineers, 2005.

7. F. Chen, P. Di, J. Huang, H. Sasaki, and T. Fukuda, *Evolutionary artificial potential field method based manipulator path planning for safe robotic assembly*, pp. 92–97. Institute of Electrical and Electronics Engineers, Nov 2009.

8. A. Badawy and C. R. Mcinnes, "Robot motion planning using hyperboloid potential functions," *World Congress on Engineering*, pp. 2–4, 2007.

9. U. Ahsun, *Dynamics and Control of Electromagnetic Satellite Formations*. PhD thesis, Massachusetts Institute of Technology, June 2007.

10. G. P. Roussos, G. Chaloulos, K. J. Kyriakopoulos, and J. Lygeros, *Control of multiple non-holonomic air vehicles under wind uncertainty using Model Predictive Control and decentralized navigation functions*, pp. 1225–1230. Institute of Electrical and Electronics Engineers, Dec 2008.

11. F. A. Gao Zhenhai and S. B. Jiang Liyong, *Optimal preview trajectory decision model of lane-keeping system with driver behavior simulation and Artificial Potential Field*, pp. 797–801. Institute of Electrical and Electronics Engineers, Jun 2009.

12. B. Ulutas, T. Haliloglu, and I. Bozma, "Folding pathways explored with articial potential functions," *Physical Biology*, vol. 6, no. 3, 2009.

13. P. Vadakkepat, K. C. Tan, and W. Ming-Liang, *Evolutionary artificial potential fields and their application in real time robot path planning*, pp. 256–263. Institute of Electrical and Electronics Engineers, 2000.

14. E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

15. A. R. Tatsch, *Artificial Potential Function Guidance for Autonomous In-Space Operations*. PhD thesis, University of Florida, 2006.

16. H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.

17. H. G. Kwatny and G. Blankenship, *Nonlinear Control and Analytical Mechanics: A Computational Approach (Control Engineering)*. Birkhauser, 2000.

18. H. Curtis, *Orbital Mechanics for Engineering Students, Second Edition (Aerospace Engineering)*. Butterworth-Heinemann, 2009.

19. J. E. Prussing and B. A. Conway, *Orbital Mechanics*. Oxford University Press, USA, 1993.

20. A. Bryson, "Optimal control-1950 to 1985," *IEEE Control Systems Magazine*, vol. 16, pp. 26–33, Jun 1996.

21. D. E. Kirk, *Optimal Control Theory: An Introduction (Dover Books on Electrical Engineering)*. Dover Publications, 2004.

22. J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, pp. 193–207, Mar 1998.

23. A. V. Rao, D. A. Benson, C. Darby, M. A. Patterson, C. Francolin, I. Sanders, and G. T. Huntington, "Algorithm 902: Gpops, a matlab software for solving multiple-phase optimal control problems using the gauss pseudospectral method," *ACM Transactions on Mathematical Software*, vol. 37, pp. 1–39, Apr 2010.

24. A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 131, pp. 497–528, 2010.

25. M. D. R. De Pinho, "A maximum principle for optimal control problems with mixed constraints," *IMA Journal of Mathematical Control and Information*, vol. 18, pp. 189–205, Jun 2001.

26. A. E. Bryson, *Dynamic Optimization*. Pearson Education, 1998.

27. D. D. Morrison, J. D. Riley, and J. F. Zancanaro, "Multiple shooting method for two-point boundary value problems," *Communications of the ACM*, vol. 5, pp. 613–614, Dec 1962.

28. D. G. Hull, "Conversion of optimal control problems into parameter optimization problems," *Journal of Guidance, Control, and Dynamics*, vol. 20, pp. 57–60, Jan 1997.

29. "Dido optimal control software: Get results fast and easy." http://www.elissarglobal.com/industry/products/software-3/, 2013.

30. "GPOPS-II: Next-generation optimal control software." www.gpops2.com, 2013.

31. J. B. Burl, *Linear Optimal Control*. Prentice Hall, 1998.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
27-03-2014	Master's Thesis	Oct 2012 - Mar 2014

4. TITLE AND SUBTITLE

Continuous Control Artificial Potential Function Methods and Optimal Control

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

R. Andrew Fields, Civ, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENY-14-M-20

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL Space Vehicles Directorate
Attn: Dr. Josue Munoz
3550 Aberdeen Avenue SE, Bldg 427
Kirtland AFB, NM 87117-5776
(505) 853-4757 (DSN: 263-4757) josue.munoz@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RV

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

Artificial potential function methods (APFMs) are a class of computationally inexpensive control methods for driving a system to a desired goal while avoiding obstacles. Although APFMs have been applied successfully to a wide range systems since the late 1980s, these control methods do have notable drawbacks. The general suboptimality of APFM results is one of these drawbacks, which is due to the fact that APFMs contain no cost function in their formulation.

This thesis first develops a new continuous control APFM for fully actuated systems called the Velocity Artificial Potential Function (VAPF) Method, which causes the system velocity to converge to the negative gradient of an artificial potential function. Then, methods for increasing APFM optimality are studied. First, an investigation is undertaken to determine if placing an APFM into an optimal control framework is a practical way of addressing the suboptimality of APFMs. While effective at increasing optimality of APFM results, this approach proves to be too computationally expensive to be practical. Finally, the Adaptive Artificial Potential Function developed by Munoz is studied and implemented via the VAPF Method. This approach produce results with higher optimality than traditional APFMs but negligibly greater computational expense.

15. SUBJECT TERMS
Artificial potential function, optimal control, obstacle avoidance

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
REPORT	**ABSTRACT**	**c. THIS PAGE**			Lt. Col Jeremy Agte (ENY)
U	**U**	**U**	**UU**	84	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636 x4667

Standard Form 298 (Rev: 8-98)
Prescribed by ANSI Std. Z39-18